Entwicklung neuer Methoden zur Modellierung technischer Systeme und zur Ergebnisauswertung für Simulationsprogramme der Reaktorsicherheit

GRS - 368

Gesellschaft für Anlagen-
und Reaktorsicherheit
(GRS) gGmbH

Entwicklung neuer Methoden zur Modellierung technischer Systeme und zur Ergebnisauswertung für Simulationsprogramme der Reaktorsicherheit

Modellierung
Simulationsprogramme

Francesco Cester
Helmuth Deitenbeck
Matthias Küntzel
Josef Scheuer
Thomas Voggenberger

April 2015

Anmerkung:

Das diesem Bericht zugrunde liegende F&E-Vorhaben RS1199 wurde im Auftrag des Bundesministeriums für Wirtschaft und Energie (BMWi) durchgeführt.

Die Arbeiten wurden von der Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH ausgeführt. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Auftragnehmer.

Der Bericht gibt die Auffassung und Meinung des Auftragnehmers wieder und muss nicht mit der Meinung des Auftraggebers

GRS - 368
ISBN 978-3-944161-49-5

Deskriptoren:

Grafische Modellierung, Simulation, Visualisierung, CAD, Computer Aided Design, ATLAS, GRAMOVIS, AGM, ATHLET GCSM Modeller, Netzwerkeditor

Kurzfassung

Das Gesamtziel des Vorhabens war die Entwicklung einer generellen Simulationsumgebung für Rechenprogrammsysteme zur Reaktorsicherheitsanalyse. Diese Simulationsumgebung stellt Methoden zur grafischen Modellierung und Ergebnisauswertung für die Simulationsmodelle bereit. Unter den Begriffen grafische Modellierung und grafische Ergebnisauswertung werden computergestützte Methoden des Prä- und Postprozessings der Simulationsmodelle zusammengefasst, die dem Anwender bei der Durchführung der Simulation Hilfestellung leisten können. Dafür eignen sich CAD („Computer Aided Design“) basierte Eingabewerkzeuge, interaktive Bedienoberflächen zur Durchführung der Simulationsrechnung sowie die grafische Aufbereitung und visuelle Darstellung der Simulationsergebnisse. Ein besonderer Schwerpunkt waren die durch den Systemcode ATHLET gestellten Anforderungen.

Es wurde ein CAD-Werkzeug entwickelt, das die Vorgabe von 3D-Geometrie der Anlage und die Diskretisierung mit einem Simulationsgitter ermöglicht. Es bietet Schnittstellen, die zur Erzeugung der Eingabedaten der Codes und zum Datenexport für das Visualisierungswerkzeug genutzt werden. Das CAD-System wurde zur Modellierung eines Kühlkreislaufs und Reaktordruckbehälters eines DWR eingesetzt.

Für die Modellbildung von komplexen Systemen mit vielen Einzelkomponenten wurde ein generell verwendbarer grafischer Netzwerkeditor adaptiert und erweitert. Dieser dient dazu, Netzwerke mit komplexer Topologie aus geeigneten Bausteinen und Elementen grafisch nachzubilden. Der Netzwerkeditor wurde zur Modellierung von Leittechnik- und Thermofluidsystemen in ATHLET erweitert und eingesetzt.

Für die visuelle Darstellung der Simulationsergebnisse im lokalen Zusammenhang mit der 3D-Geometrie und dem Simulationsgitter wurde das Open-Source-Programm ParaView verwendet, das für die 3D-Visualisierung von Felddaten weit verbreitet ist und umfangreiche Optionen zur Darstellung und Auswertung der Daten bereitstellt. Mit den entwickelten Methoden können die Ergebnisse aus den RS-Codes sowie die Daten aus der CAD-Modellierung konvertiert und anschließend in ParaView importiert und visualisiert werden. In der Anwendung ergeben sich Vorteile bei Analysen mit sehr detaillierter Diskretisierung. Dies gilt besonders für die Analyse von Phänomenen mit starken, lokalen Unterschieden und die Validierung mehrdimensionaler Modelle.

Abstract

The overall objective of the project is to develop a general simulation environment for program systems used in reactor safety analysis. The simulation environment provides methods for graphical modeling and evaluation of results for the simulation models. The terms of graphical modeling and evaluation of results summarize computerized methods of pre- and postprocessing for the simulation models, which can assist the user in the execution of the simulation steps. The methods comprise CAD ("Computer Aided Design") based input tools, interactive user interfaces for the execution of the simulation and the graphical representation and visualization of the simulation results. A particular focus was set on the requirements of the system code ATHLET.

A CAD tool was developed that allows the specification of 3D geometry of the plant components and the discretization with a simulation grid. The system provides interfaces to generate the input data of the codes and to export the data for the visualization software. The CAD system was applied for the modeling of a cooling circuit and reactor pressure vessel of a PWR.

For the modeling of complex systems with many components, a general purpose graphical network editor was adapted and expanded. The editor is able to simulate networks with complex topology graphically by suitable building blocks. The network editor has been enhanced and adapted to the modeling of balance of plant and thermal fluid systems in ATHLET.

For the visual display of the simulation results in the local context of the 3D geometry and the simulation grid, the open source program ParaView is applied, which is widely used for 3D visualization of field data, offering multiple options for displaying and analyzing the data. New methods were developed, that allow the necessary conversion of the results of the reactor safety codes and the data of the CAD models. The transformed data may then be imported into ParaView and visualized. The application of 3D visualization shows advantages in transient analyses with very detailed discretization. This is especially true for the analysis of phenomena with strong local differences and the validation of multi-dimensional models.

Inhaltsverzeichnis

1 Einleitung

Die Anforderungen an Genauigkeit und Aussagesicherheit von Rechenprogrammsystemen zur Simulation von Stör- und Unfallabläufen in Reaktorkern, Kühlkreislauf und Sicherheitsbehälter von KKW haben sich in den letzten Jahren ständig erhöht. Die Analyse von Sicherheitsfragestellungen mit stark lokalen Einflüssen und Effekten, wie etwa Deborierungsereignisse, thermische Belastungen des Reaktordruckbehälters, Rekritikalität oder Filmsieden im Reaktorkern, erfordert Simulationen mit hoher räumlicher Auflösung, beispielsweise durch Systemcodes mit sehr detaillierter Systemmodellierung, in einigen Fällen auch gekoppelte Analysen mit mehrdimensionaler Neutronendynamik. Mit der Leistungsfähigkeit und dem Umfang der Simulationsprogramme wächst auch ihre Komplexität, die für den Anwender erhöhte Anforderungen bei der fachgerechten Erstellung der notwendigen Eingabedaten für die Modelle und der Auswertung der berechneten Daten bedeuten. Deshalb werden fortschrittliche, computergestützte Methoden benötigt, die ihn bei der Vorbereitung, der Durchführung und Auswertung der Analysen unterstützen. Diese Methoden helfen, Fehler bei der Datenerstellung zu vermeiden und das Verständnis der Ergebnisse von Simulationen mit auftretenden Phänomenen zu verbessern. Die vorhandenen Werkzeuge zum Prä- und Postprozessing der Reaktorsicherheitscodes, wie beispielsweise codespezifische Prüfverfahren der Eingabedaten und die 2D-Visualisierung der Ergebnisdaten mit ATLAS werden damit entsprechend der neuen Anforderungen ergänzt.

2 Zielsetzung

Ziel der Arbeiten war es, eine möglichst generelle Simulationsumgebung für grafische Modellierung und Ergebnisauswertung bereitzustellen, die für mehrdimensionale Simulationsmodelle der Thermohydraulik und Neutronendynamik verwendet werden kann. Unter den Begriffen grafische Modellierung und grafische Ergebnisauswertung werden computergestützte Methoden des Prä- und Postprozessings der Simulationsmodelle zusammengefasst, die alle für eine Simulation notwendigen Schritte unterstützen. Für die vorgesehene Anwendung sind das insbesondere CAD („Computer Aided Design“) basierte Werkzeuge zur Vorgabe von Geometrie und Gitterdaten, grafische Verfahren zur Nachbildung thermohydraulischer und leittechnischer Systeme, Prozeduren zur Erzeugung der modellspezifischen Eingabeformate sowie die Auswertung und Aufbereitung der Simulationsergebnisse und ihre visuelle Darstellung im lokalen Zusammenhang und zeitlichen Ablauf.

Der Einsatz von computerbasierten Methoden trägt wesentlich zur Qualität der Analysen bei. Mit den grafischen Eingabesystemen können z. B. Herstellerunterlagen und Daten mit größerer Präzision und geringerer Fehlerwahrscheinlichkeit in ein Computermodell übertragen werden. Gleichzeitig sollen die Daten in einer weitgehend unbearbeiteten Form in den Systemen gespeichert werden und können damit als Referenzdaten für die modellierten Komponenten und Systeme immer wieder genutzt werden. Zusätzlich sollen durch eine automatisierte Umsetzung der Darstellungen in den Eingabesystemen für das Visualisierungssystem äquivalente Darstellungen erzeugt werden können, die mit den Simulationsdaten verknüpft sind.

Wesentliche Aufgaben der Datenvisualisierung sind die Überprüfung und das Verständnis des räumlichen Zusammenhangs und der zeitlichen Entwicklung von Simulationsergebnissen, beispielsweise bei der Erkennung lokaler Strömungsphänomene. Zusätzlich können die Ergebnisse mit der Visualisierung in anschaulicher Form nach außen präsentiert werden. Die Möglichkeiten der etablierten 3D-Visualisierungssysteme sind heute so vielfältig, dass sich fast jede gewünschte Sicht auf die Daten realisieren lässt. Die Ergebnisse können z. B. als Konturgrafen, Iso-Oberflächen, Vektorfelder, Partikelwolken, farbige Volumina mit Licht und Transparenzeffekten dargestellt werden. Die Anzeige eines Drahtgitters oder von Komponentendetails kann optional ergänzt werden. Hinzu kommen viele interaktive Optionen wie Wahl des Blickwinkels, Schnitte oder die Darstellung verdeckter Volumina. Neben der Auswahl

eines geeigneten Visualisierungssystems sollte im Vorhaben eine möglichst generelle Schnittstelle für den Import der Geometrie-, Gitter- und Ergebnisdaten der Modelle bereitgestellt werden. Außerdem ist es sinnvoll, aus den vielfältigen Visualisierungsoptionen diejenigen auszuwählen, die für die Darstellung der Ergebnisse der verwendeten Simulationsmodelle besonders geeignet sind. Die Anwendung des Visualisierungssystems kann zusätzlich durch entsprechende Empfehlungen, spezieller Dokumentation und der Automatisierung von Arbeitsabläufen erleichtert werden, mit dem Ziel, den Aufwand für den Anwender zu vermindern.

Eine elementare Anforderung an die eingesetzte Software war, dass Funktionen von Entwicklern selbst geändert oder ergänzt werden können und die entwickelten Tools mit den Simulationsmodellen kostenfrei für die Anwender verfügbar sind. Dies wird erfüllt, wenn die entwickelten Methoden und Werkzeuge auf OpenSource Software aufsetzen, die einen breiten Entwickler- und Anwenderkreis hat und damit eine kontinuierliche Weiterentwicklung sicherstellt. Als Alternativen kommen der Kauf entsprechender Rechte in kommerzieller Software oder vollständige Eigenentwicklung in Frage.

3 Arbeitsprogramm

Im Vorhaben wurden zur Erfüllung der Zielsetzung die folgenden Arbeitspakete (AP) festgelegt und bearbeitet: Alle Arbeiten konnten wie geplant umgesetzt werden und damit auch die Ziele des Vorhabens erreicht werden.

AP 1: Grafische Modellierung

Die Arbeiten zielen darauf ab, die in den Simulationsmodellen benötigten Daten interaktiv zu erzeugen. Ein Werkzeug auf CAD-Basis soll entwickelt werden, mit dem die Geometrie der Anlage abgebildet werden kann und das neben den allgemein vorhandenen Standardfunktionen Schnittstellen für verbreitete Geometrieformate und Programmierbarkeit aufweist. Diese Schnittstellen werden genutzt, um in den Geometriemodellen die benötigten Gitter für die Modelle, die Nodalisierung, festzulegen. Zusätzliche Modelldaten, wie Materialdaten, Wirkungsquerschnittsdaten, Reibungsbeiwerte sollen ebenfalls erfasst, gespeichert und dokumentiert werden können. Das Gesamtsystem und die entwickelten Methoden sollen an einem Modell eines Reaktordruckbehälters (RDB) getestet werden.

Für die Modellerstellung von komplexen leittechnischen oder thermohydraulischen Reaktorsystemen soll ein generell verwendbarer grafischer Netzwerkeditor entwickelt oder adaptiert werden. Dieser soll die Nachbildung der Topologie der Netze durch das interaktive Zusammenschalten von Objekten mit Verbindungselementen ermöglichen. Der interaktive Editor soll eine erweiterbare Objektbibliothek mit Verbindungselementen, einen Objektbrowser, Eingabemasken für Objektdaten und die notwendige Datenspeicherung umfassen. Systemtests sollen mit den erzeugten ATHLET Eingabedaten und einer systeminternen Simulation direkt durchführbar sein.

Die Anwendung des Editors für das ATHLET-Leittechnikmodell, dem „General Control System Module (GCSM)“ /AUS 90/ ist vorgesehen und die vorhandenen Leittechnikblöcke werden dort implementiert. Die interaktive Modellierung, die Simulation und die Eingabedatenerzeugung werden mit Beispielen von leittechnischen Systemen getestet.

AP 2: Methoden zur Auswertung und Visualisierung von Simulationen

Ein Softwarewerkzeug zur Darstellung und Auswertung von Felddaten mehrdimensionaler Simulationen soll für die besonderen Anforderungen der Reaktorsicherheitscodes

adaptiert werden. Insbesondere sollen die Programme ATHLET, QUABOX/CUBBOX und TORT zur Simulation der thermohydraulischen und neutronenphysikalischen Vorgängen im Reaktor unterstützt werden. Methoden zur Anzeige zeitdynamischer Daten, für die Generierung abgeleiteter Daten, die Nutzung von Datenfiltern und eine Anzeige lokaler Daten als Zahlenwert oder Achsendiagramm sollen neben den mehrdimensionalen Darstellungen verfügbar sein. Die mit dem CAD-System generierten Geometrie und Nodalisierungsdaten sollen automatisiert übernommen und mit den lokalen Ergebnisdaten verknüpft werden können.

Ein einheitliches, standardisiertes Datenformat für alle benötigten Daten soll verwendet werden, das die Speicherung und den Zugriff auf räumliche und zeitliche Ergebnisdaten, Geometrie- und Nodalisierungsdaten sowie Zusatzinformationen ermöglicht.

Die Bedienung des Visualisierungssystems soll für den Benutzer möglichst komfortabel und einfach durchführbar sein und wenig erfahrenen Benutzern den Einsatz erleichtern. Daher werden geeignete Darstellungsoptionen untersucht und dokumentiert, die die modellspezifischen Bedürfnisse der betrachteten Simulationsmodelle berücksichtigen. Die gewählten Einstellungen und Optionen sollen gespeichert werden, damit die Darstellungen reproduziert oder für andere Simulationen zur Verfügung gestellt werden können. Zum Test des Gesamtsystems soll ein Modell eines Reaktordruckbehälters erstellt werden, das die Daten aus einer mehrdimensionalen Störfallsimulation visualisiert.

4 Stand von Wissenschaft und Technik

Im Folgenden wird die Ausgangssituation des Stands der Technik für das Prä- und Postprozessing wichtiger Simulationsmodelle für die Thermohydraulik und Neutronendynamik beschrieben. Dabei wird auf Modelle für integrale Berechnungen und Systemcodes, Neutronendynamikmodelle sowie CFD-Modelle für mehrdimensionale Strömungsberechnungen näher eingegangen.

Systemcodes

Für die Systemsimulation von Transienten und Störfällen wird in der GRS der thermohydraulische Systemcode ATHLET /ATH 09/ eingesetzt und weiterentwickelt. Für die Simulation der Vorgänge im Containment kommt das gekoppelte Programmsystem COCOSYS, für die Simulation auslegungsüberschreitender Störfälle der Integralcode ASTEC zum Einsatz. Mit diesen Rechenprogrammen wird eine realistische Simulation der Störfallabläufe in Reaktoranlagen angestrebt. Sie nutzen zur örtlichen Diskretisierung meist ein im Vergleich zu den CFD-Modellen relativ grobes Gitter und die Lösungen werden nur in Hauptströmungsrichtung berechnet. Dennoch wird durch verfeinerte Gitter und die Verwendung gekoppelter Einzelcodes die Auflösung der dargestellten Geometrie immer komplexer. Die Daten für die einzelnen Komponenten der Modelle („Objekte“) müssen in Form von Eingabedaten in Dateien entsprechend einem vorgegeben Format aus den Herstellerunterlagen erstellt werden. Die zunehmende Simulationstiefe, also z. B. örtliche Auflösung, Redundanzen und detaillierte Leittechnikmodellierung, führen zu sehr großen Datensätzen mit extrem vielen Einzelobjekten. Diese lassen sich letztlich nur noch durch funktionale Tests (Beispielrechnungen) mit verschiedenen Anwendungsfällen testen. Die Erstellung der Datensätze wird bereits teilweise mit interaktiven, grafischen Werkzeugen unterstützt. Ebenfalls existieren automatisierte Verfahren zur Prüfung, Auswertung und grafischen Darstellung der Eingabedaten.

Für ATHLET gibt es eine grafische Benutzeroberfläche, mit der aus vorliegenden Isometrieplänen Daten für Leitungen, Krümmer, Abzweige etc. eingeben werden können und diese in die benötigten Objektdaten von ATHLET konvertiert werden. Es ist eine Reihe von Standardberechnungen enthalten, die z. B. Längskoordinaten, die Rohrreibung und Wärmeleitzahlen berechnen können. Der komplette Eingabeumfang für ATHLET-Thermohydraulikobjekte wird aber nicht unterstützt. Diese Software basiert

auf der heute nur noch wenig genutzten Programmiersprache „SMALLTALK“ und lässt sich wegen der Nutzung fremdentwickelter Softwaremodule nicht erweitern. Für das Leitechniksimulationsmodell GCSM /AUS 90/ aus ATHLET ist ebenfalls grafischer Editor vorhanden, der zur Nachbildung der leittechnischen Systeme im Rahmen der Erstellung von Anlagensimulatoren der deutschen KKW für den BMU genutzt wird. Der GCSM-Eingabegenerator /JAK 97/ ist mit der Expertensystemsoftware „G2“ von GENSYM realisiert. Dies ist ein sehr mächtiges Softwarewerkzeug, z. B. für Prozessüberwachung, das sehr hohe Lizenzkosten erfordert und deshalb nicht allgemein verfügbar sein kann.

Für die Ergebnisauswertung von ATHLET /ATH 09, Vol. 1/ werden eine Vielzahl verschiedener Ansätze genutzt. Die meisten werden als sogenannte „Utilities“ mit ATHLET installiert und bieten neben alphanumerischen Auswertungsverfahren verschiedene Programme zur Erzeugung von X/Y-Diagrammen, also Darstellung der Ergebnisgrößen als Funktion der Zeit. Ein Standardwerkzeug für diesen Zweck ist das kostenfreie Programm „APTPLOT“ /APT 15/, das in Java implementiert und für UNIX und WINDOWS verfügbar ist. Die größte Flexibilität zur Visualisierung bietet ATLAS /VOG 10/, das auf Anforderung allen Nutzern von ATHLET zur Verfügung gestellt wird. Neben der Datenvisualisierung in Achsendiagrammen kommen 2D-Darstellungen hinzu, mit denen die Ergebnisse in Form von Prozessbildern anschaulich in Form veränderlicher Farben oder Formen angezeigt werden können. Einige Darstellungen, wie Thermohydraulik- und Wärmeleitobjekte, GCSM Signalverknüpfungen, Brennstabvisualisierung können durch Auswertung der Eingabedaten automatisch erzeugt werden und erfordern nur geringen Aufwand zur Erzeugung für den Benutzer. Zusätzlich besteht die Möglichkeit, die Simulation während des Ablaufs („Online“) interaktiv zu beeinflussen, um beispielsweise Ventile zu steuern oder Ausfälle und Störungen zu simulieren.

Das Programmsystem COCOSYS /ALL 05/ dient der Simulation der Thermohydraulik und des Aerosolverhaltens im Sicherheitsbehälter und angrenzenden Gebäuden während Stör- und Unfällen. Zusätzlich existiert eine gekoppelte Version zum Programm ATHLET-CD, um die Zustände und Rückwirkungen im Reaktorkühlkreislauf zu berücksichtigen. Für das Prä- und Postprozessing werden in angepasster Form ähnliche Werkzeuge genutzt, die bereits bei ATHLET beschrieben wurden. Dies gilt für einen interaktiven Eingabeeditor mit den notwendigen Eingabemasken auf Basis von SMALLTALK, eine automatische Bildgenerierung für die Nutzung zur Ergebnisvisualisierung in ATLAS und für die X/Y-Diagrammsoftware COCPLOT. Die Erzeugung der

Eingabedaten für die Räume (Zonen) für hohe örtliche Auflösung (mehrere 100) kann mit Hilfe des Gittergenerators „GRIDGEN“ durchgeführt werden. GRIDGEN ist eine kommerzielle Software zur Erzeugung eines Diskretisierungsgitters der Fa. Pointwise /POI 15/. Das Werkzeuge importiert entweder vorliegende Geometrien aus CAD-Systemen oder erlaubt es, die Geometriedaten direkt mit Punkten, Kanten und Flächen mit den Verbindungen zu definieren. Dieses Verfahren wird in COCOSYS verwendet und die definierten Objekte werden zu den vorgesehenen Zonen für die Simulation manuell zusammengefasst. Diese Informationen werden in einem Interfaceprogramm ausgewertet und damit die COCOSYS-Eingabedaten erstellt. Die von GRIDGEN bereitgestellten Geometriedaten können außerdem in ein für das 3D-Visualisierungsprogramm FIELDVIEW /INT 15/ lesbares Format umgewandelt werden. Da die Zonennamen bereits in GRIDGEN definiert sind, lassen sich darüber die dynamischen Ergebnisdaten aus COCOSYS direkt und automatisch dem Gitter in FIELDVIEW zuordnen.

Die international entwickelten System- und Integralcodes TRACE, CATHARE, MELCOR und ASTEC werden mit eigenen Werkzeugen für Prä- und Postprozessing ausgeliefert. Außerdem sind ASTEC und MELCOR auch an ATLAS gekoppelt und können alle dort vorhandenen Visualisierungsmethoden nutzen. Zusätzlich wurde für die ASTEC Module „CESAR“ (Thermohydraulik) und „DIVA“ (Brenn- und Steuerstäbe) die ASTEC Input Grafic (ASIG) in der GRS entwickelt, mit der sowohl die Eingabedaten grafisch überprüft werden als auch Bilder zur Visualisierung der Ergebnisse in ATLAS erzeugt werden können.

Für die US-Codes TRACE, CONTAIN, COBRA, FRAPCON-3, MELCOR, PARCS und RELAP ist das, „Symbolic Nuclear Analysis Package“ (SNAP) /APT 07/ verfügbar, das für Erstellung der Eingabedaten, Ausführung und Auswertung der Analysen verwendet werden kann. SNAP ist als generelles Werkzeug so konzipiert, dass eine Erweiterung auf zusätzliche Codes möglich ist und bietet Schnittstellen, um für diese Codes sogenannte SNAP-Plugins zu entwickeln. Wesentliche Elemente von SNAP sind der Model Editor und Navigator. Im Editor können einzelne Komponenten des Modells, z. B. Fills, Pipes, Pumps usw. grafisch in einem Netzwerk zu einem Anlagenmodell zusammengesetzt werden. Im Navigator werden die erzeugten Komponenten in einer Baumstruktur angezeigt und die Dialoge zur Eingabe der Komponentendaten können aufgerufen werden. Neben den physikalischen Daten können auch sogenannte „Views“ der Komponenten definiert werden, die für die Visualisierung der dynamischen Daten und zu interaktiven Steuerung verwendet werden. Die Möglichkeiten der Datenanzeige sind

ähnlich wie die in ATLAS vorhandenen, also generell 2D-Grafiken und einfache 3D-Darstellungen für ein kartesisches, regelmäßiges Gitter.

Modelle der Neutronendynamik

Betrachtet werden hier die für Reaktorsicherheitsanalysen in Deutschland verwendeten Modelle DYN3D /FZD 15/ und QUABOX/CUBOX /LAN 05/ auf Basis der Diffusionsgleichungen mit relativ groben Gittern sowie die Transportmodelle DORT-TD und TORT-TD /SEU 08, CHR 10/ für sehr feine Gitter. Im Rahmen der Integration von Reaktorsicherheitsmodellen in der europäischen Simulationsplattform NURESIM/NURISP wurde DYN3D mit Software-Bausteinen aus SALOME /SAL 15/ in die Plattform integriert /KLI 09/. SALOME ist ein OpenSource Produkt der französischen Fa. Open Cascade S.A., Tochter von EURIWARE/AREVA. Es ist für verschiedene LINUX-Derivate verfügbar und ist generell zur Kopplung numerischer Simulationsmodelle vorgesehen. Es umfasst u.a. Werkzeuge zur CAD Modellierung („Geometry“), Gittergenerierung („Meshing“) und Visualisierung („Post-Pro/VISU“). Die Kommunikation der einzelnen Module untereinander setzt auf einem Industriestandard zur Prozesskommunikation „CORBA“ auf, die Daten können mit Hilfe von Bibliotheksfunktionen im einem standardisierten Format (HDF5) in Dateien gespeichert und ausgetauscht werden. Die CAD-Komponente erlaubt die Übernahme von Daten aus anderen CAD-Systemen oder bietet die Möglichkeit, die Geometrie aus Basisobjekten und üblichen CAD-Funktionen interaktiv zu erstellen. Der Gittergenerator in SALOME basiert auf NETGEN, einer österreichischen OpenSource Software zur automatisierten Erstellung von Tetraedergittern aus CAD-Modellen. Andere kommerzielle Gittergeneratoren sind vorhanden oder können über entsprechende Schnittstellen integriert werden. Das generierte Gitter kann z. B. als Drahtmodell visualisiert werden und zur weiteren Verwendung in Dateien gespeichert werden. Benötigte zusätzliche Funktionen und Bedienoberflächen können mit den in SALOME vorhandenen Programmierschnittstellen in der Sprache PYTHON ergänzt werden. Das Visualisierungsmodul nutzt das „Visualization Toolkit VTK“, eine OpenSource C++ Bibliothek für Bildverarbeitung und Visualisierung. Damit sind sowohl 2D als auch 3D-Darstellungen der Geometrie, des Gitters und der Ergebnisdaten möglich. Die Ergebnisse können beispielsweise farbcodiert als Skalar- und Vektorfelder, in Schnittebenen oder als Isosurfaces angezeigt werden. Die Integration von DYN3D in NURESIM mit SALOME ist weit fortgeschritten und daher sind viele Optionen bereits verfügbar. Mit dem Geometrie- und Meshing-Werkzeug kann ein Reaktorkern mit der gewünschten Nodalisierung erzeugt werden und in weiteren Dialogen zusätzliche Daten wie Wirkungsquerschnitte zugeordnet wer-

den. Daraus können dann die für DYN3D notwendigen Eingabedateien erzeugt werden und die Simulation direkt aus SALOME- Bedienoberfläche durchgeführt werden. Die Visualisierung der Ergebnisse kann sowohl während der Rechnung als auch aus den Ergebnisdateien erfolgen.

Für die Gittererzeugung und Eingabegenerierung von QUABOX/CUBBOX und TORT-TD werden bisher keine interaktiven Werkzeuge verwendet.

Für die Visualisierung der Ergebnisse von QUABOX/CUBBOX werden die kommerziellen Produkte FIELDVIEW und AVS Express /AVS 15/ verwendet, für die Datenschnittstellen zu den Ergebnisdaten vorhanden sind. Speziell für die gekoppelte Simulation mit ATHLET wurde eine 2D-Visualisierung in ATLAS implementiert, in der radiale Schnitte durch den Reaktorkern und axiale Leistungsprofile dargestellt werden. Die radialen Zonen und axialen Ebenen in den Diagrammen werden automatisch aus den ATHLET Eingabedaten generiert, die die Zuordnung von Leitungsobjekten bzw. Fluidkanälen zum Gitter der Brennelemente enthält. Für die Programme DORT/TORT können die Ergebnisdaten durch Import in Microsoft Excel oder speziell entwickelte 2D-Bilder ausgewertet und dargestellt werden.

CFD-Modelle für Fluid- und Strömungsdynamik

Für die Analyse von Störfällen mit lokalen physikalischen Vorgängen ist die Modellierung von Vermischungsvorgängen mit mehrdimensionalen Ansätzen notwendig. Zur Bereitstellung eines mehrdimensionalen Fluidmodells in ATHLET ist das Rechenprogramm CFX der ANSYS-Germany für zweiphasige Strömungen erweitert sowie mit ATHLET gekoppelt /PAP 09/. CFX bietet im Rahmen der ANSYS-Workbench viele interaktive Werkzeuge zur Unterstützung der Simulation. Die Workbench enthält einen Geometrie-Designer, in den auch Daten aus andern CAD-Programmen importiert werden können. Darauf aufbauend ist ein Gittergenerator vorhanden, um sowohl regelmäßige als auch unstrukturierte Gitter zu erzeugen. Dieses Werkzeug wird durch einen konfigurierbaren Präprozessor für zusätzliche Daten (Randbedingungen, Physik) ergänzt und somit können alle notwendigen Daten für eine CFD-Analyse interaktiv generiert werden. Außerdem enthält die Workbench auch einen Postprozessor, mit dem die Analyseergebnisse der CFX-Rechnung mit der Darstellung von Geometrie und Gitter visualisiert werden können. Die dynamischen Anzeigen können direkt als Animation exportiert werden. Bisher existieren allerdings noch keine gemeinsamen interaktiven Methoden für die Durchführung und Auswertung einer gekoppelten Rechnung mit

ATHLET. Das Programm COBRA-TF /PER 10/ ist ein thermohydraulisches Unterkanalmodell für den Reaktorkern mit 3-D Feldgleichungen für 2-phasige Strömungen. Das Programm wurde in Zusammenarbeit mit der Pennsylvania State University an QUABOX/CUBBOX sowie TORT-TD gekoppelt. Die oben erwähnte Schnittstelle zu SNAP ist für diese Version von COBRA nicht verfügbar, so dass bisher neben einfachen Microsoft Excel Diagrammen 3D-Darstellungen mit dem OpenSource Tool „VisIt“ für grafische Analysen des Lawrence Livermore National Laboratory /LLN 15/ verwendet wurden.

5 Grafische Modellierung

Die im Folgenden vorgestellten Arbeiten stellen Methoden bereit, die die Erstellung codespezifischer Eingabedaten durch interaktive, grafische Werkzeuge unterstützen. Ein Ziel war es, ein Werkzeug zu Verfügung zu stellen, mit dem die Geometrie der Anlage in modellunabhängiger Weise abgebildet werden kann und zusätzliche Modelldaten, insbesondere für die Diskretisierung, zu erfassen. Zur Erstellung dreidimensionaler Computermodelle von Anlagenkomponenten, z. B. Brennelement, Reaktorkern, Druckbehälter ist die Nutzung eines CAD–Systems geeignet. Die beabsichtigte erweiterte Anwendung erfordert Schnittstellen für unterschiedliche Datenformate und Programmierbarkeit des Systems. Das Gesamtsystem und die entwickelten Erweiterungen wurden mit den 3D-Modellen eines RDB und den daran anschließenden Kühlmittelleitungen getestet.

In einem weiteren Arbeitsschwerpunkt wurde für die Modellbildung von komplexen Systemen mit vielen Einzelkomponenten ein generell verwendbarer grafischer Netzwerkeditor adaptiert. Bei der Modellierung ist ein grafisches Werkzeug von großem Nutzen, da mit diesem die Systemmodelle in Analogie mit vorliegenden Papierplänen am Bildschirm erstellt werden können. Bei komplizierten Systemen erleichtert dieses Verfahren ein System toplogisch korrekt zu modellieren. Der Netzwerkeditor wurde für die Eingabedatengenerierung von ATHLET für Leittechnik- und Thermohydrauliksysteme beispielhaft ausgebaut. Damit können aus den vorhandenen Funktionsblöcken interaktiv Modelle generiert werden, notwendige Parameter vorgegeben und die ATHLET Eingabedaten erzeugt werden. Für Leittechniksysteme ist eine dynamische Simulation direkt im Netzwerkeditor möglich.

5.1 Vergleich und Test von CAD-Werkzeugen zur Komponentenmodellierung

Im ersten Schritt wurden verschiedene CAD-Programme daraufhin getestet, damit eine präzise, grafische 3D Darstellung zu erstellen. Außerdem wurden prinzipielle Eigenschaften für die Konstruktion sowie der Preis und die Verfügbarkeit verglichen.

Insgesamt wurden 5 Programme näher analysiert: FreeCAD, NaroCAD, TurboCAD, SketchUp von Google und der ANSYS DesignModeler. Die beiden ersten sind OpenSource Entwicklungen, wohingegen die anderen als kommerzielle Produkte ver-

trieben werden. Das ANSYS Programm wurde wegen der Verfügbarkeit in der GRS zum Vergleich herangezogen. Es wird darauf hingewiesen, dass die folgenden Aussagen auf den Stand der Entwicklung der Programme zum Beginn des Vorhabens (2011) bezogen sind und daher Abweichungen zum aktuellen Zustand nicht berücksichtigen.

Mit jedem von diesen CAD-Programmen wurde versucht, ein vereinfachtes 3D-Model des RDB (Zylinder mit 2 Halbkugeln) mit verschiedenen Komponenten wie Leitungen, Stutzen, etc. zu zeichnen. Besonders geprüft wurde, wie einfach die Konstruktion nachträglich angepasst werden kann, z. B. die Geometrien. Zusätzlich wurden wichtige Eigenschaften an Hand der von Beschreibungen der Programme ermittelt. Die Ergebnisse des ausführlichen Vergleichs sind im Anhang 1 dargestellt.

SketchUp ist eine Software, die zur Erstellung von 3D-Gebäudemodellen für Google Earth optimiert wurde. Die Bedienbarkeit ist für Laien zwar sehr leicht erlernbar, aber die exakte Vorgabe und Änderung geometrischer Parameter ist in der Konstruktion nur sehr aufwändig oder gar nicht möglich.

TurboCAD ist eine preiswerte, weit verbreitete Konstruktionssoftware. Von der Funktionalität zeigt der Vergleich leichte Vorteile gegenüber den OpenSource-Programmen, aber die fehlende Anpassbarkeit durch eine Skript-Schnittstelle beschränkt die geplante Erweiterung.

Aus diesen Gründen blieben nach der Vorauswahl lediglich FreeCAD und NaroCAD übrig und es war die Frage zu klären, auf welches CAD-System besser aufgebaut werden kann, um die Anforderungen für die geplanten Erweiterungen mit möglichst geringem Aufwand zu erfüllen.

Da beide Systeme auf der Technologie von Open CASCADE /OCT 15/ aufbauen sind sie in ihren Möglichkeiten zur Konstruktion praktisch identisch. Auch die Bedienbarkeit der Systeme war in den getesteten Versionen in etwa vergleichbar. Details sind in der folgenden Tab. 5.1 aufgeführt.

Tab. 5.1 Funktionen von FreeCAD und NaroCAD im Vergleich

Funktion	Free CAD	NaroCAD
Maße eingeben	Ja	Ja
Maße als Parameter	Ja	nein
Aufwand beim Messen	Niedrig	Niedrig
Extrudieren	Ja	Ja
Schnitt	?	Ja
Rotation	Ja	Ja
Vol. Erstellen bei shape+guidingLine	nein	Ja
Parallele Ebene nach XYZ erstellen	Ja	Ja
Nicht parallele Ebene erstellen	nein	?
Ebene auf einem Körper erstellen	nein	nein
Queransicht auf eine Ebene	nein	nein
Skizze Modus (2D)	Ja	Ja
Skizze Modus (2D) auf einem Körper	?	nein
Aufwand Skizze 2D	Niedrig	Sehr niedrig
Matrix Funktion	nein	Ja
Spiegelfunktion 2D	nein	Ja
Spiegelfunktion 3D	Ja	Ja
Intersektion	nein	Ja
Subtraktion	nein	Ja
Python Scripting	ja	nein
Formate	.igs/.stp/.mcnp/etc.	.naroxml

Als ausschlaggebende Kriterien wurden daher folgende Eigenschaften herangezogen:

- Leichte Anpassbarkeit/Erweiterbarkeit
- Datenformate für Export/Import
- Plattformunabhängigkeit
- Weiterentwicklung und Entwickler-Community

In allen dieser Kategorien konnte FreeCAD, mit zum Teil deutlichem Vorsprung punkten, wodurch die Entscheidung letztendlich zugunsten von FreeCAD /FRC 15/ als Basissystem ausfiel.

Die Möglichkeiten FreeCAD anzupassen und zu erweitern sind sehr vielfältig. Praktisch alle Funktionen zur geometrischen Modellierung sind über den eingebetteten Python-Interpreter zugänglich. Darüber hinaus kann dieser auch auf die Module zur Erzeugung und Steuerung der grafischen Benutzeroberfläche zugreifen. Daraus ergibt sich ein sehr breites Spektrum an Möglichkeiten zur Erweiterung durch Python-Module. Somit ist auch eine Entwicklung per „rapid prototyping“ sowohl für geometrische Konstruktion als auch für GUI-Elemente problemlos möglich. Auch zeitkritische Berechnungen, die in C/C++ implementiert werden, können sehr einfach als Python-Modul DLLs voll integriert werden.

Für NaroCAD, besteht zwar auch die Möglichkeit Abläufe in der Skriptsprache Lua zu beschreiben. Allerdings war die System-Einbettung des Interpreters in der getesteten Version deutlich weniger ausgereift. Als Datenformate für den Export und Import von Geometrie stehen in FreeCAD, im Gegensatz zu NaroCAD, bereits eine Reihe gängiger Formate zur Verfügung. Dadurch besteht bereits eine gute Anbindungsmöglichkeit zu den geplanten Anwendungen zur Netzwerkbearbeitung und der Modell-Visualisierung.

Da für die zukünftige Nutzung des Anwendungspakets die typische Plattform nicht direkt absehbar ist, ist auch die Plattformunabhängigkeit ein interessanter Aspekt. FreeCAD wurde bereits als plattformunabhängiges Projekt konzipiert und basiert auch ausschließlich auf entsprechend breit verfügbaren 3rd-Party Bibliotheken. Durch die reine C++-Implementierung der nativen Programmteile ist für FreeCAD eine native Compilation auf einer Reihe von Windows- und Unix-Systemen möglich. NaroCAD setzt hier im Gegenzug auf C# und die .Net-Plattform, wodurch eine plattformübergreifende Nutzung zwar theoretisch angedacht, in der Praxis allerdings unrealistisch ist. Auch sind Kompatibilitätsprobleme durch die gleichzeitige interne Nutzung von OpenGL und DirectX in Zukunft nicht auszuschließen.

Letzter betrachteter Punkt war die zukünftig zu erwartende Weiterentwicklung des Grundsystems durch dessen Entwickler. Eine rege Community von Nutzern und Entwicklern ist Voraussetzung um von Fehlerbereinigungen und neuentwickelten Funktionen für das Basissystem profitieren zu können. Auch hier schneidet FreeCAD im Vergleich zu NaroCAD deutlich besser ab, da in den entsprechenden Foren ein viel regerer Austausch stattfindet und dort auch Fragen zu Entwicklungsdetails erfahrungsgemäß schnell beantwortet werden.

5.2 Entwicklung von Funktionen zur Generierung von Eingabedaten für mehrdimensionale Simulationsmodelle aus einem CAD-System

Die Entwicklung der grafischen Modellierung wurde auf Basis des OpenSource Projekts FreeCAD realisiert. Bis auf wenige Erweiterungen in dem als C++ Applikation entwickelten Kern konnten durch den integrierten Interpreter sämtliche Bestandteile in Python realisiert werden. Diese Tatsache sorgte nicht nur für die gewissen Erleichterungen bei der Entwicklung des Prototyps sondern trug durch die Flexibilität Pythons auch dazu bei, die Weiterentwicklungen des Kerns durch die Community immer wieder ohne große Anpassungen übernehmen zu können. Diese Abstraktion erwies sich als sehr wertvoll, besonders da viele Funktionen des Kerns unter ständiger Entwicklung standen. Grundsätzlich verfügt FreeCAD bereits in der offiziellen Release-Version über alle Funktionen um komplexe Anlagen zu konstruieren. Um zum gewünschten Ergebnis zu kommen müssen diese Funktionen allerdings auch mit der entsprechenden CAD-Erfahrung bedient werden. Bei den meisten Anwendern der Codes kann diese Erfahrung aber im Allgemeinen nicht vorausgesetzt werden. Hinzu kommt, dass die Konstruktionsmechanismen, wie in jedem CAD-System, sehr generisch gehalten sind um in ihrer Anwendung die größtmögliche Bandbreite abzudecken. Für eine konkrete Aufgabenstellung, wie z. B. die Konstruktion eines Rohrleitungsstranges, bedeutet das, alle notwendigen Parameter zu spezifizieren sind, selbst wenn diese konstant sind oder in direkter Beziehung zueinander stehen und so automatisch berechnet werden könnten. Durch die Festlegung der vorrangig zu unterstützenden Konstruktionsaufgabe, nämlich die Modellierung von Rohrleitungsverläufen, konnten die Ziele für die entwickelte FreeCAD-Erweiterung wie folgt definiert werden:

- einfache Überführung von Isometrieplänen in 3D-Rohrleitungskonstruktionen
- Fehlervermeidung durch hohen Grad an Automatisierung von häufig wiederkehrenden Konstruktionsschritten
- unmittelbare Fehlerkontrolle durch sofortige Umsetzung von Konstruktionsschritten in 3D-Geometrie
- Möglichkeit zur Gruppierung von Konstruktionsschritten durch Makrobauteile.

5.2.1 Konzept und Realisierung

Die 3D-Modellierung von Anlagenteilen geschieht mit der Zielsetzung, die erstellte Geometrieinformation sowohl für die Visualisierung von ATHLET-Ergebnissen als auch zur Überführung in ATHLET-Eingabedaten nutzen zu können. Für beide Anwendungen ist es notwendig die Geometrieobjekte mit der ATHLET-eigenen Aufteilung und Organisation in Thermofluid-Objekte (TFOs) und deren Einteilung entlang ihrer Länge in Nodes (Nodalisierung) bereitzustellen. Neben den geometrischen Randbedingungen muss hierfür auch die Nomenklatur erstellter TFOs eingehalten werden. Für die Umsetzung von Isometrieplänen in 3D-Konstruktionen ist die von ATHLET propagierte Art der Modelldefinition allerdings nicht besonders hilfreich, da diese einige für die 3D-Modellierung wichtige Parameter, wie Winkel und Bauteilorientierung, nicht berücksichtigt. Darüber hinaus sind zuweilen Angaben notwendig, wie z. B. Nodegrenzen oder die Zuordnung von Bauteilen zu TFOs, die dem Benutzer zum Zeitpunkt der rein geometrischen Konstruktion noch gar nicht vorliegen. Aus diesen Gründen schien es sinnvoll, die Umsetzung von Plänen in 3D-Modelle in zwei Phasen einzuteilen - die geometrische Konstruktion und die Festlegung der ATHLET-spezifischen Objektorganisation.

5.2.1.1 Geometrische Konstruktion

In der ersten Phase konzentriert sich der Benutzer auf die Konstruktion der Geometrie. Mit Hilfe der Anlagenpläne sollten die hierfür notwendigen Angaben, wie Längen, Winkel und Durchmesser, einfach gemacht werden können. Besonders dadurch, dass komplexe Rohrverläufe meist als Isometriepläne vorliegen, sollten die entwickelten Funktionen die isometriebasierte Konstruktion bestmöglich unterstützen. Hieraus ergab sich, die Funktionen der FreeCAD-Erweiterung auf die schrittweise Eingabe und eine sukzessive Erstellung der Geometrie hin zu optimieren.

Eine sehr wichtige Aufgabe der entwickelten Funktionen ist es, dem Benutzer Koordinatenberechnungen weitestgehend abzunehmen und so die Konstruktion zu erleichtern. Das hierzu entwickelte Konstruktionswerkzeug, eine sog. PipeTurtle, erlaubt es, die aus den Plänen ermittelten Daten, wie z. B. Längen, Winkel und Rohrweiten direkt zu verwenden und die Rohrverläufe der Anlage Schritt für Schritt nachzubilden. Frei nach dem Prinzip der Turtle Graphik nimmt das Werkzeug diese Informationen als Befehlssequenzen entgegen und konstruiert daraus die 3D-Geometrie-Segmente der entsprechenden Rohrleitungen, indem es Schnittprofile entlang der vorher berechneten

Mittellinie extrudiert. Die Konstruktion erfolgt dabei nach einem zustandsbasierten Verfahren, welches alle dafür notwendigen Geometrieberechnungen übernimmt. Damit werden aus einfachen Längenangaben die entsprechenden Rohrstücke der aktuellen Nennweite konstruiert und direkt an den bisherigen Rohrverlauf angefügt. Winkel werden hierbei automatisch als Bögen und Änderungen der Nennweite als Reduzierstücke gemäß der aktuell eingestellten Verjüngungsrate realisiert. Die PipeTurtle ist mit einem stapelbasierten Speicher ausgestattet, womit der aktuelle Zustand, inklusive Position und Orientierung, gesichert und später wiederhergestellt werden kann. Dadurch ist es auch sehr einfach möglich Abzweigungen zu konstruieren.

Das Ergebnis einer einfachen Befehlssequenz ist in Abb. 5.1 dargestellt, welche die grundsätzliche Funktionsweise des Werkzeugs zeigt. Wie in der Abbildung ersichtlich wird, werden Position und Orientierung der PipeTurtle durch Liniensegmente symbolisiert, wodurch sich die nächsten Schritte besser planen lassen. Im unteren Bereich des Fensters ist die Python-Konsole eingeblendet, die in diesem Beispiel dazu benutzt wurde um die zur Konstruktion nötige Befehlssequenz direkt in Form von Python-Funktionsaufrufen einzugeben. In aller Regel werden Anwender diese Schritte zwar per Point&Click über die Funktionsleiste oder das Menü umsetzen. Interessant bleibt diese Art der Konstruktion dennoch, da sich hiermit den geübten Anwendern sämtliche Möglichkeiten zur Automatisierung und zur Erstellung eigener Makrobauteile eröffnen.

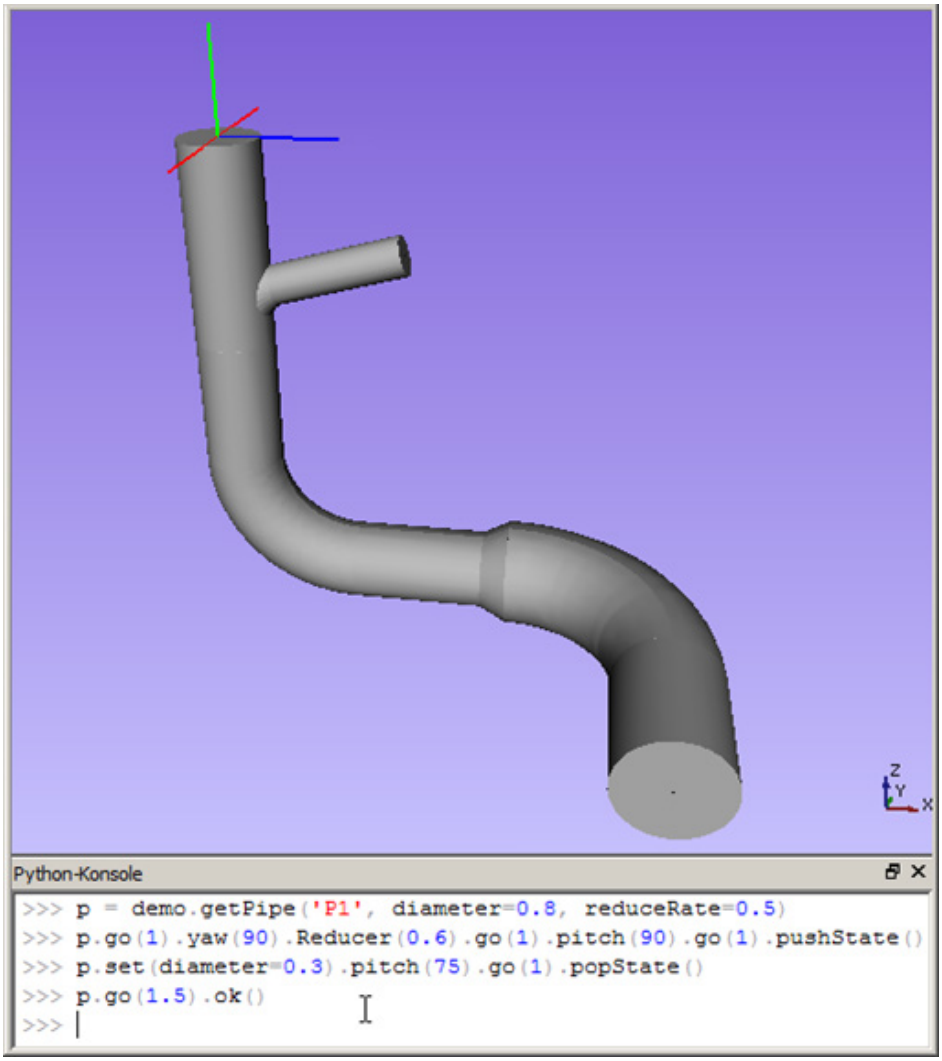


Abb. 5.1 Konstruktion durch Python-Konsole

Aus diesen einfachen Befehlssequenzen lassen sich auch komplexere geometrische Elemente zusammensetzen. Für die typischen Armaturen Reduzierungen, Ventile und Pumpen, wie sie in Leitungsnetzwerken immer wieder gebraucht werden, wurden diese bereits als parametrisierbare Makrobauteile zusammengefasst und können genau wie einfache Konstruktionsschritte als Befehl aufgerufen und passend platziert werden. Abb. 5.2 zeigt dies am Beispiel von Ventilen und einer Pumpe. Durch die Realisierung als Makrobauteil besitzen die Armaturen die Fähigkeit sich automatisch an den Kontext anzupassen in dem sie verwendet werden. Im Falle einer nachträglichen Änderung der Rohrweite ändern sich damit auch alle im weiteren Verlauf platzierten Armaturen.

Die so erzeugten Geometrie-Segmente ergeben bereits ein rohes 3D-Modell, welches schon während der Konstruktion aus verschiedenen Richtungen betrachtet und überprüft werden kann. Geometrische Unstimmigkeiten, z. B. Überschneidungen oder un-

passende Rohrweiten, die durch fehlerhafte Parameterangaben entstehen, werden damit sofort ersichtlich und sind entsprechend korrigierbar.

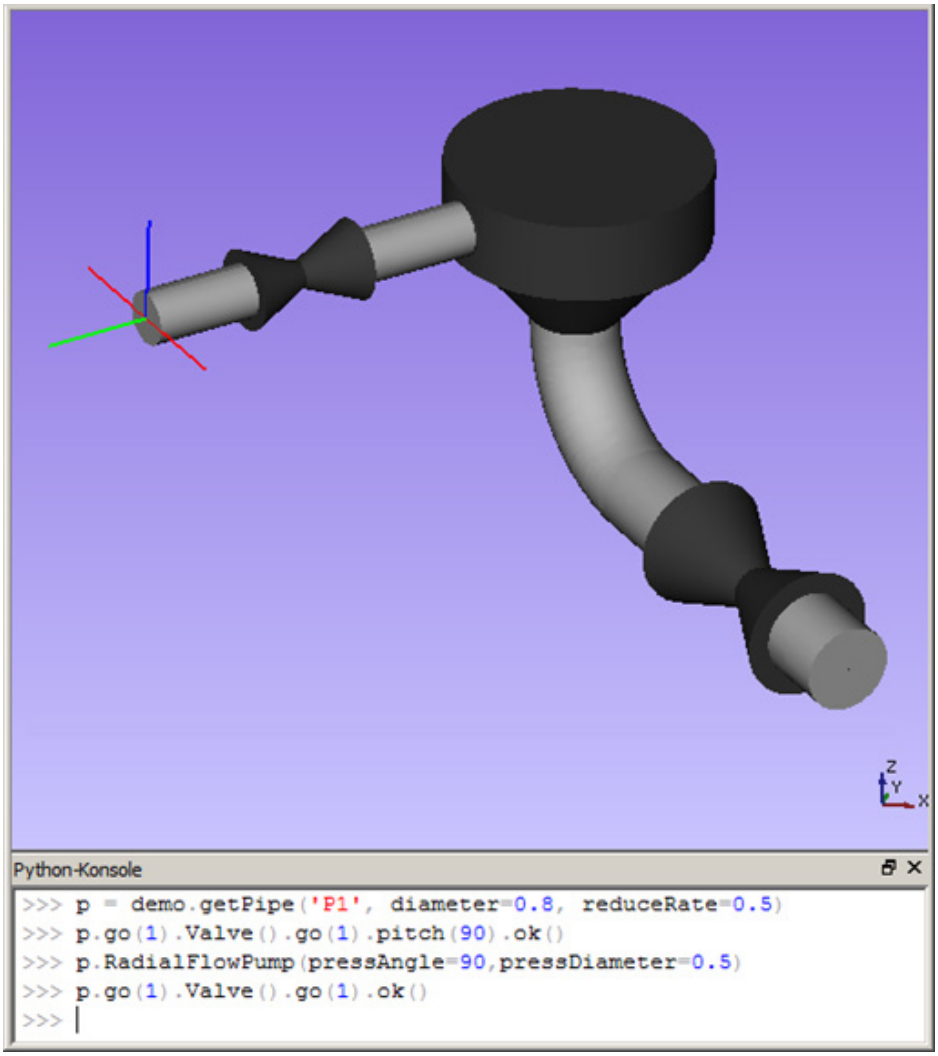


Abb. 5.2 Adaptive Armaturen als Makrobauteile

5.2.1.2 ATHLET-spezifische Einteilung des Modells

Um das in der ersten Phase durch geometrische Befehlssequenzen konstruierte Anlagenmodell ATHLET-konform zu diskretisieren, müssen die erstellten Leitungsteile den TFOs zugeordnet und geeignet nodalisiert werden. Für die TFO-Zuordnung geometrischer Objekte reicht es aus die ATHLET-Nomenklatur einzuhalten und diese mit einem entsprechenden Namenspräfix zu versehen. Schwieriger wird es allerdings, die ATHLET-Nodalisierung entlang der abgewickelten Länge eines TFOs umzusetzen. Ein TFO kann dadurch aus einer beliebigen Anzahl von ATHLET-Nodes bestehen, die für

die Visualisierung als eigenständige Geometrieobjekte vorliegen müssen. Im Allgemeinen setzen sich diese Nodes aus Teilstücken der geometrischen Leitungsteile zusammen, was bedeutet, dass letztere an den spezifizierten Nodegrenzen unterteilt werden müssen. Der flexible und unregelmäßige Verlauf von Rohrleitungen macht es unmöglich, die Leitungsteile im Nachhinein durch Boolsche Operationen aufzuteilen. Dadurch bleibt nur die Option, die Nodegrenzen gleich bei der Erstellung der Leitungsgeometrie zu berücksichtigen. Die PipeTurtle wurde hierfür mit der Fähigkeit ausgestattet, optional eine vorgegebene Nodalisierung einzuhalten und beim Erreichen einer Nodegrenze das aktuell generierte Geometriesegment stückweise zu extrudieren. Aus diesem Grund veranlasst, genau wie die Änderung geometrischer Parameter, auch die TFO-Zuordnung und Nodalisierung die PipeTurtle dazu, den Leitungsstrang neu zu erstellen.

Abb. 5.3 zeigt das Beispiel eines Rohrleitungsstranges dessen Konstruktionsbefehle dem TFO_0 zugeordnet wurden. Die Zuordnung der Befehle wird auch im Dokumentbaum (links) ersichtlich. Jeder Befehlstyp wurde zur besseren Identifizierbarkeit mit einem entsprechenden Icon ausgestattet. Die Befehlsknoten selbst geben in ihrer Reihenfolge im Baum die der Reihe nach ausgeführten Befehle an, die die PipeTurtle zur Erzeugung des TFOs abarbeitet. Die zu den Befehlen zugehörigen Parameter, wie Längen und Winkel, können nach Auswahl eines Befehls im Eigenschaftsfenster (links unten) angepasst werden. Veränderungen der Befehlsparameter werden automatisch registriert und führen zur Neuerstellung des betroffenen TFOs.

Im Beispiel wurde der Trace-Knoten des ersten TFOs (TFO_0) ausgewählt und die Nodalisierung im Eigenschaftsfenster angepasst. Hierzu wurde eine Reihe von Längen spezifiziert, die an den entsprechenden Positionen für Nodegrenzen sorgen. Um diese in der Darstellung besser überprüfen zu können, kann, wie in der Abbildung ersichtlich, die Transparenz der Rohrgeometrie erhöht werden. Zusätzlich dazu wurden an den Schnittpositionen Marker eingeblendet, die die abgewickelte TFO-Länge an den Schnittpunkten abgeben und diese als Nodegrenzen kennzeichnen.

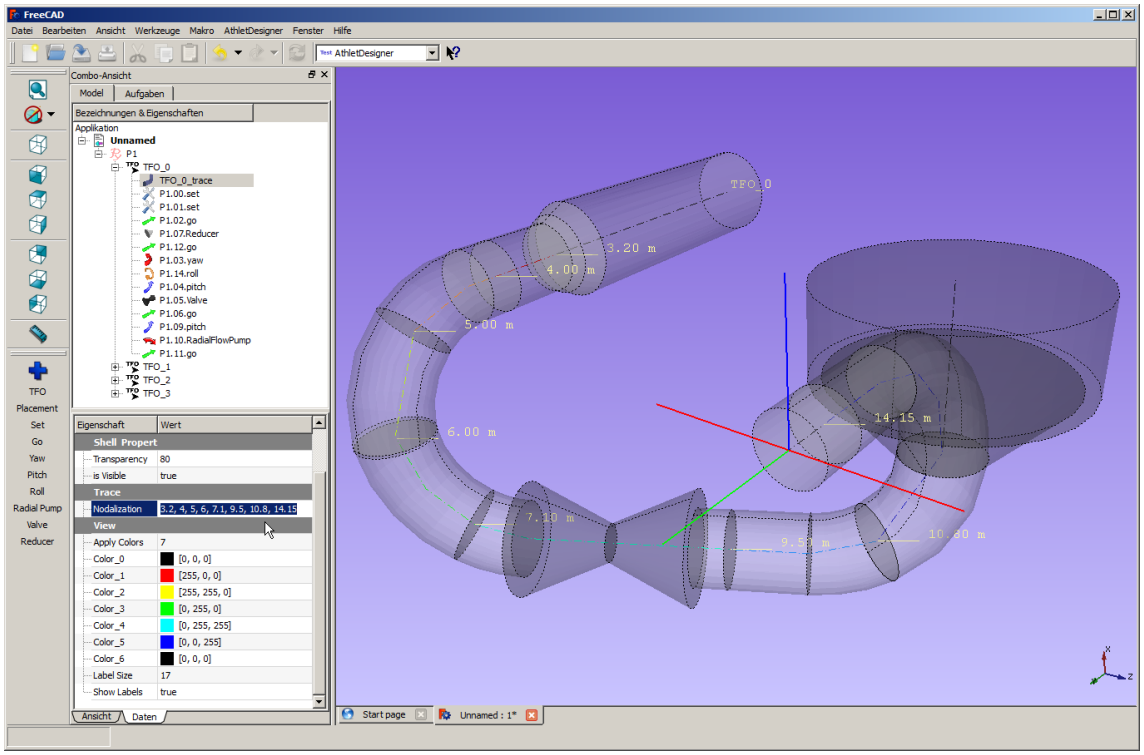


Abb. 5.3 Interaktive Nodalisierung von TFOs

5.2.2 Geometrie-Export

Um das in FreeCAD erstellte Anlagenmodell für die Visualisierung verwendbar zu machen, muss die Geometrie der erzeugten AHTLET-Nodes richtig benannt, in Meshes umgewandelt und exportiert werden. Als Datenformat wird hierbei das weit verbreitete Wavefront OBJ genutzt, welches FreeCAD auch schon in der offiziellen Version schreiben und lesen kann. Allerdings bieten die nativen Export- und Import-Filter des OBJ-Formates bislang keine Unterstützung für einige wichtige Features. Die Meshes geometrischer Objekte werden dadurch lediglich als lose Sammlung von Facetten exportiert - sowohl die Gruppierung der Facetten zu Objekten als auch Namenszuordnungen gehen dabei verloren. Hierfür mussten erweiterte Export- und Import-Filter implementiert werden, die diese Mängel beheben und darüber hinaus erlauben, mit zusätzlichen Metadaten in den OBJ-Dateien umzugehen. Dadurch können jetzt auch OBJ-Dateien eingelesen werden, die von anderen Programmen, wie z. B. Blender oder Google SketchUp, erstellt wurden.

5.2.3 Erstellung von Modellen eines Reaktors für einen Testfall

Mit den Anlagenplänen und Isometrien für einen DWR lagen die wichtigsten Maße der Anlagengeometrie vor und konnten mit Hilfe des entwickelten Werkzeugs recht einfach in FreeCAD übertragen werden. Die Einteilung der Rohrgeometrie in nodalisierte TFOs erfolgte dabei entsprechend den Vorgaben eines bestehenden ATHLET-Datensatzes. Aufgrund der Symmetrie reichte es aus nur den ersten Loop der Anlage detailliert zu modellieren und diesen parametrisiert als Makrobauteil zusammenzufassen.

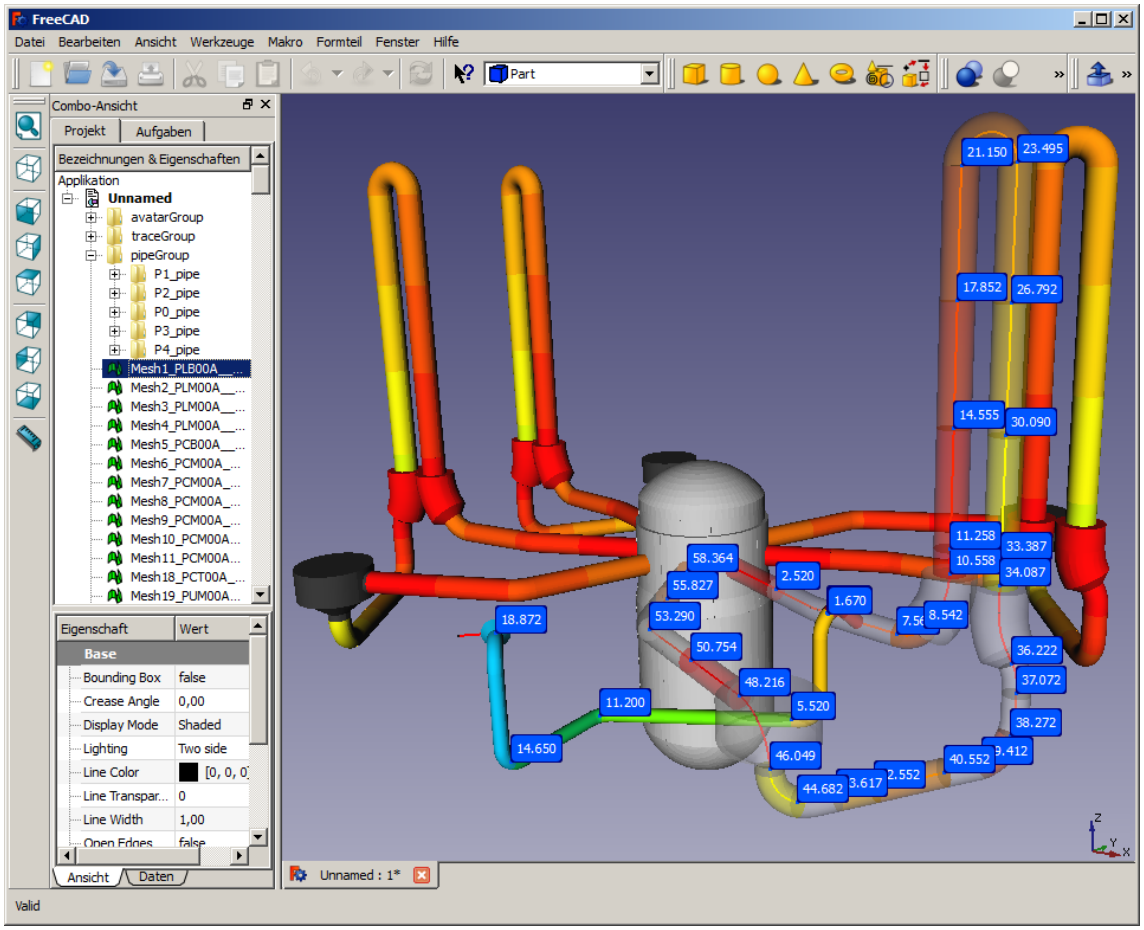


Abb. 5.4 Konstruktion und Nodalisierung für Testszenario GKN2

Dieses wurde dann verwendet, um alle Primärkreise zu erstellen (Abb. 5.4). Auch die Möglichkeit eines funktionierenden OBJ-Imports erwies sich hier als sehr wertvoll. Ein bereits existierendes, aber in SketchUp erstelltes Modell des Reaktordruckbehälters konnte dadurch in das Rohrleitungsmodell integriert und für die Visualisierung eines Pumpenausfallszenarios (siehe Kap. 6.5) genutzt werden.

5.2.4 Erzeugung von RDB-Geometrie

Ein weiteres Ziel des Projektes war das Ausloten der Möglichkeiten um RDB-Geometrie direkt in FreeCAD zu konstruieren. Ein Vorgehen, das dazu getestet wurde, ist die Erstellung eines parametrisierten Makrobauteils auf Basis des für die Rohrleitungsgeometrie entwickelten Werkzeugs. Dieser Ansatz folgte aus der Überlegung, dass RDB-Geometrie in größtenteils regelmäßige, vertikale verlaufende Kanäle aufgeteilt werden kann, welche dann als Leitung mit annähernd trapezförmigem Querschnitt extrudiert werden können. Als positiver Nebeneffekt werden bei diesem Verfahren auch gleich die Mittellinien berechnet, die den in der Visualisierung gebrauchten Hauptströmungsrichtungen entsprechen. Die dafür notwendigen Erweiterungen in die PipeTurtle einzubauen, stellte kein größeres Problem dar. Allerdings stellte sich dieses Vorgehen aus anderen Gründen als nicht praktikabel heraus und wurde daher nicht weiter verfolgt:

- Trotz hoher Symmetrie gibt es in RDB-Geometrien eine sehr große Anzahl an Parametern die in der Modellierung einstellbar sein sollten. Hieraus ergibt sich die Gefahr von widersprüchlichen Parameterangaben, die zu invalider Geometrie führen können. Deshalb sollte der Benutzer bei der Angabe der Parameter unterstützt werden und nicht alle spezifizieren müssen oder auch können. Diese Ansprüche alle innerhalb eines Makrobauteils abzudecken stellte sich als sehr schwierig heraus und die angestrebte intuitive Bedienung des Werkzeugs dürfte nur schwer zu erreichen sein.
- Besonders im Bereich der Kalotte ergibt sich bei Erstellung der Nodes sehr hoher Aufwand, da typischerweise einige Unstetigkeiten in der Aufteilung zu berücksichtigen sind. Dennoch sollten keine Lücken in der Geometrie entstehen, die dann in der Visualisierung als Löcher in Erscheinung treten würden.
- FreeCAD stößt bisher auf massive Performanceprobleme falls viele über den Dokumentbaum verwaltete Knoten, wie z. B. die PipeTurtle-Befehle, erstellt werden. RDB-Geometrien bestehen bereits bei konservativen Nodalisierungen aus 500 Nodes und mehr und würden entsprechend viele Befehlsknoten erfordern. Die zu erwartenden Zeiten für Geometrieaktualisierungen wären beachtlich und würden eine interaktive Parametereingabe nahezu unmöglich machen.
- Es gibt einige Einschränkungen in der Konstruktion da die Extrusion von Leitungsprofilen ist nicht immer fehlerfrei ist. Dadurch werden Fehler, wie Löcher oder Singularitäten in exportierten Geometriemeshes begünstigt, die in der weiteren Verarbeitung zu Problemen führen.

Da sich der oben beschriebene Ansatz als nicht zielführend herausstellte, wurde versucht ein RDB-Modell mit den in FreeCAD nativ vorhandenen Konstruktionsmethoden aufzubauen, was sich aufgrund der beachtlichen Weiterentwicklungen in FreeCAD vergleichsweise problemlos realisieren ließ. Hierzu wurden vor allem Techniken wie das Extrudieren von 2D-Skizzen, Rotationsgeometrien und Boolsche Operationen genutzt. Zusätzlich war es bei der Erstellung und Parametrisierung der Modellkomponenten sehr hilfreich, die Skripting-Möglichkeiten mittels Python nutzen zu können.
Im Besonderen konnten regelmäßige und symmetrische Teile des RDB-Modells durch die in FreeCAD eingebauten Array-Befehle einfach realisiert werden. Diese Befehle zeichnen sich dadurch aus, dass einzelne Objekte gemäß eingestellter Parameter vervielfacht (geklont) werden und die daraus resultierende Menge an Objekten (Array) als Gesammtheit weiter verarbeitet werden kann. Die hierfür einstellbaren Parameter umfassen z. B. die Kopieanzahl, Anordnungsart (orthogonal oder radial), und Richtungsvektoren für den Versatz und können nachträglich verändert werden. Das hiermit erstellte RDB-Modell bleibt durch diese Parameter also in einem gewissen Rahmen flexibel und kann an veränderte Bedingungen angepasst werden. Möglichkeiten zur Automatisierung boten sich hier dadurch, die Parameter dieser Array-Objekte per Python-Skript anzupassen.

Insbesondere die richtige Benennung der resultierenden Nodes, deren Namen sich meist aus TFO-Bezeichnung, Sektor- oder Kanalnummer und Index zusammensetzen, stellte eine Herausforderung dar. Erschwert wird dieses auch dadurch, dass die regelmäßige Indizierung der Nodes in der ATHLET-Modellierung nicht immer eingehalten wird. Um die Nodenamen richtig bestimmen zu können, wurde ein halbautomatisches Verfahren implementiert, welches auch Verbundobjekte (Arrays) korrekt behandeln kann. Es erlaubt pro Geometrieelement Namensregeln festzulegen, die dann beim OBJ-Export alle Nodeobjekte als getrennte Meshes verarbeitet und gemäß den festgelegten Namensregeln eindeutig benennt. Dieses Vorgehen ist auch deshalb notwendig, da das OBJ-Format keine Objekt-Hierarchie und somit auch keine strukturellen Zugehörigkeiten kennt, was eine Namenskonvention bedingt, um die eindeutige Zuordenbarkeit von Nodemeshes zu ihren TFOs sicherzustellen.

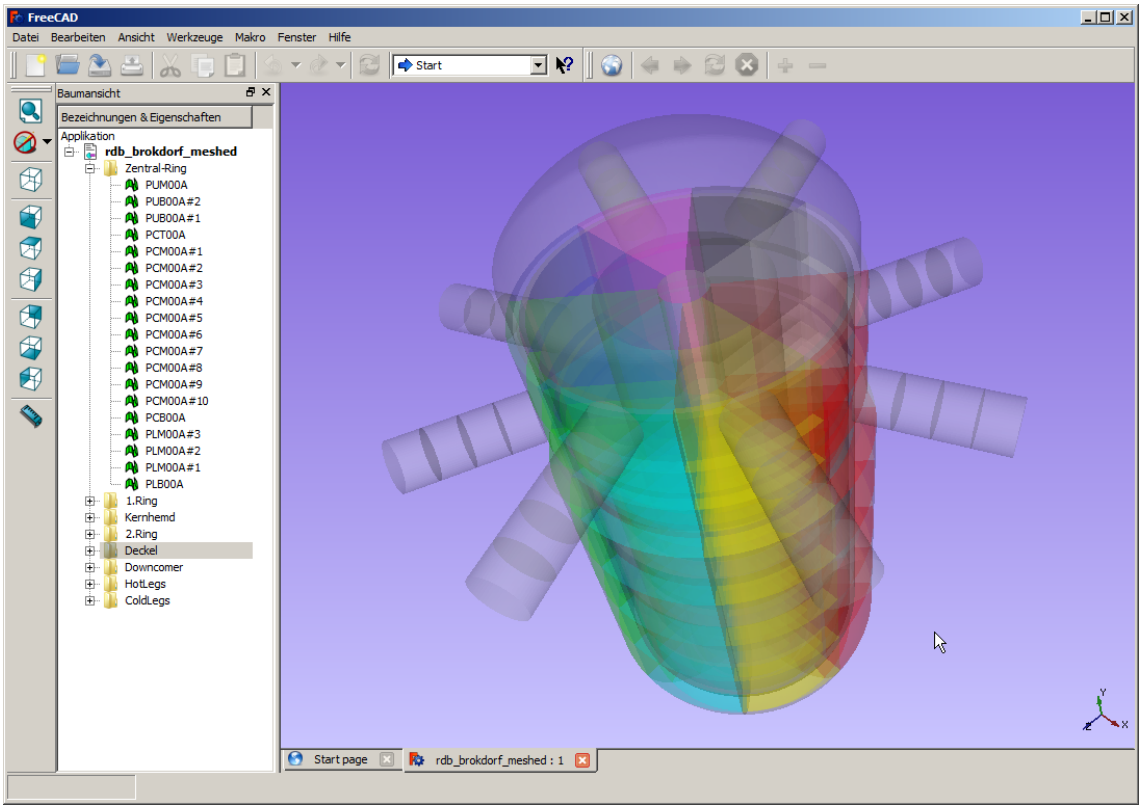


Abb. 5.5 Diskretisierung (Mesh) eines RDB-Modell in FreeCAD

Abb. 5.5 zeigt das Ergebnis des nativ konstruierten RDB-Modells nachdem es als OBJ exportiert und zurück in FreeCAD importiert wurde. Wie das Bild zeigt wurden die Nodes beim Export entsprechend der AHTLET-Konventionen benannt und die Hierarchie des ursprünglichen Dokuments erhalten. Nach dem Reimport wurden die Node-Meshes zwecks besserer Überprüfbarkeit auf eine halbtransparente Darstellung gesetzt und entsprechend ihrer Sektoren koloriert.

5.3 Entwicklung eines grafischen Netzwerkeditors

Zur Systemsimulation von thermohydraulischen und leittechnischen Systemen mit Systemcodes wie ATHLET oder COCOSYS werden interaktive Eingabegeneratoren benötigt, mit denen man grafisch die benötigten Modelle erstellen kann. Dafür müssen Netzwerke mit komplexer Topologie aus geeigneten Bausteinen und Elementen nachgebildet werden. Dazu werden erweiterbare Bibliotheken von Basiselementen benötigt, die man auf einer Zeichenebene zu dem Gesamtmodell verbinden kann. Für die einzelnen Elemente müssen Parameter (Eingabedaten) in Benutzerdialogen vorgegeben

werden können. Zusätzlich werden Masken für allgemeine codespezifische Daten benötigt (z. B. Priority chains, numerische Daten, Subsystem Definition in GCSM etc.).

5.3.1 Auswahl der Basissoftware

Für die interaktive Erstellung solcher Netzwerkmodelle sollte ein generischer grafischer Netzwerkeditor entwickelt oder adaptiert werden, der für unterschiedliche Simulationsprogramme anpassbar und erweiterbar ist und die folgenden wesentlichen Anforderungen erfüllt:

- Erweiterbare Bausteinbibliothek
- Zusammenschaltung der Elemente mit Verbindungen zwischen Ein- und Ausgängen
- Einfache Anpassung der Eingabemasken für die Parameter
- Datenprüfung
- Testmöglichkeit der Systeme mit Randbedingungen
- Erzeugung codespezifischer Eingabe oder von Quellcodeprogrammen
- Kostenfrei für die Anwendung verfügbar
- Verfügbarkeit des Quellcodes und Programmierschnittstellen

Die Neuentwicklung eines solchen Netzwerkeditors ist sehr aufwändig. Daher wurden existierende Lösungen auf ihre Eignung untersucht. Im kommerziellen Bereich gibt es eine Reihe guter Ansätze, wie z. B. „Simulink“, eine Blockdiagrammumgebung für die Mehrdomänen-Simulation und Model-Based Design /MAT 15/. Dieses kann zwar als Referenz für Design und Funktionalität dienen, kommt aber wegen der Kosten und der Nichtverfügbarkeit des Quellcodes für eine Erweiterung nicht in Frage. Ein vergleichbares, ebenfalls kommerzielles System, wird vom russischen Hersteller 3V Services unter der Bezeichnung „SimInTech“ angeboten. Es dient generell zur blockorientierten Modellierung und Simulation technischer Systeme /SIT15A/. Das Hauptanwendungsgebiet ist das Design und die Analyse komplexer Prozessregelsysteme. SimInTech wird, in modifizierter Form, u.a. zur Modellierung von WWER-Anlagen bei einem russischen Simulator-Hersteller verwendet wird. Im Gegensatz zu MathWorks bietet der Hersteller auch den Kauf des Quellcodes von SimInTech an, mit dem Recht, damit erstellte An-

wendung „runtime free“, also ohne zusätzliche Gebühren, an andere Anwender weiterzugeben. Damit erfüllt das Produkt wichtige Voraussetzungen für die im Vorhaben geplante Anwendung.

Im Bereich von frei zugänglicher Software konnte im Hinblick auf die geforderten technischen Möglichkeiten nur das SCICOS/XCOS Programmpaket, das ähnlich wie Simulink, zur grafischen Modellierung und Simulation dynamischer Systeme entwickelt wurde. Es ist eine interaktive Simulationsumgebung für SCILAB, das ursprünglich von INRIA als Alternative zu MATLAB in Frankreich entwickelt wurde /SCI 15/. XCOS ist eine auf Java basierende Weiterentwicklung von SCICOS, die neben SCILAB auch andere Simulationssoftware, wie z. B. Modelica, unterstützt. Die Entwicklung wird seit 2012 unter der Regie von „Scilab Enterprises“, einem Konsortium verschiedener Industrieunternehmen, als OpenSource Projekt durchgeführt.

Im ersten Arbeitsschritt wurden SimInTech und Scilab in einem intensiven Vergleich bewertet. Die Softwareprodukte wurden dazu zur Evaluierung installiert und nach einer Einarbeitungsphase umfangreichen Tests unterzogen. Als Referenz wurde der bisher in ATHLET verwendete G2 basierte GCSM Eingabegenerator verwendet. Die Simulation eines einfachen Anwendungsbeispiels aus der Regelungstechnik wurde, soweit möglich, in allen Systemen durchgeführt, besonders auch im auf die Bedienbarkeit und die Benutzerschnittstellen.

Die getesteten Systeme sind prinzipiell geeignet, die grundsätzlichen Anforderungen zu erfüllen und die bisher verwendete Lösung zu ersetzen, stellen also umfangreiche Bibliotheken von Basiselementen zur Verfügung, die grafisch in der Oberfläche verknüpft werden können. Die Simulation des erstellten Systems ist dann direkt aus der Bedienoberfläche möglich, einschließlich von Ergebnisanzeigen in Zeitdiagrammen. Beide Werkzeuge waren in der Lage, das Testsystem zufriedenstellend zu simulieren. Beim Bedienkonzept bestehen jedoch erhebliche Unterschiede, die Scilab-Oberfläche XCOS entspricht nicht dem heute üblichen Standard für Bedienoberflächen (GUI). Ähnliches gilt bei der internen Durchführung der Simulation und der Ergebnisauswertung. SimInTech ist zwar nur für WINDOWS verfügbar, bietet dort aber ähnlichen Bedienkomfort wie andere aktuelle Anwendungssoftware.

Als weitere Entscheidungsgrundlage wurde die Erzeugung benutzerspezifische Blöcke getestet. Aus vorhandenen Basiselementen können in XCOS eigene Blöcke abgeleitet werden und beispielsweise die Ein- und Ausgänge, die Icons und die Funktionen in

Fortran, C und in der XCOS-Sprache definiert werden. Eine Entwicklungsumgebung steht allerdings nicht zur Verfügung. Auch die Gestaltung neuer Benutzerdialoge und Menüs ist unter Änderung von XCOS-Unterprogrammen zum Menüaufbau oder mit Scilab Funktionen zur GUI-Erzeugung möglich. Eine völlige Neugestaltung der Benutzerschnittstelle, wie sie für eine moderne, zeitgemäße Bedienung benötigt würde, wäre allerdings sehr zeitaufwändig.

SimInTech bietet verschiedene Möglichkeiten, eigene Simulationsblöcke zu ergänzen. Die interne Modellierung kann mit einer internen Simulationssprache oder der Einbindung einer externen Funktionsbibliothek erfolgen. Zusätzlich zur internen Systemsimulation enthält SimInTech Werkzeuge zur Generierung von C-Programmcode und den entsprechenden Programmbibliotheken. Damit können modellierte Systeme auch in anderen Umgebungen verwendet werden. Auch die Definition eigener Symbole für die Blöcke und die Erstellung der Parametereingabe sind sehr effizient, ohne Programmierung, realisierbar. Alternativ ist auch die direkte Änderung der SimInTech Source in der modernen Entwicklungsumgebung „DELPHI/XE“ /EMB 15/ möglich. Dort ist auch eine beliebige Anpassung der Bedienoberfläche möglich.

Beide Systeme erlauben die Speicherung der erstellten Simulationsdiagramme mit allen zugeordneten Daten in binärer und textueller Form. Bei SimInTech wird die ASCII- Datei im XML-Format erzeugt, das einfach und teilweise selbsterklärend interpretiert werden kann. Dies erleichtert die Erstellung eines Konverters zum Import aus anderen Simulationssystemen oder zu Geometriemodellen aus FreeCAD erheblich.

Die Möglichkeit, externe Daten einzuspielen und anzuzeigen ist von ebenfalls von Interesse, wenn man das System zur Visualisierung einer externen Simulation, wie beispielsweise mit ATHLET, nutzen will. Bei SimInTech sind entsprechende Schnittstellen vorgesehen, die die Entwicklung dieser Funktionalität erleichtern, während der Aufwand bei XCOS wesentlich höher eingeschätzt werden muss.

Zusätzlich wurden eine Reihe weiterer Funktionalitäten geprüft die für die geplanten Einsatz von Bedeutung sind, wie z. B. Simulationsdurchführung, Darstellung der Ergebnisse in Diagrammen, Strukturierung und Navigation in großen Systemen sowie die Integration von Online-Hilfe. In diesen Bereichen waren die Basissysteme weitgehend gleichwertig.

Insgesamt wurden die Eignung und die Anpassungsfähigkeit beim Netzwerksystem SimInTech besser bewertet als bei XCOS. Insbesondere der verfügbare Support des Herstellers bei der Erweiterung des Systems war ein weiteres Argument zu Gunsten dieser Basissoftware.

5.3.2 Grundfunktionen des Netzwerkeditors

In den folgenden Abschnitten werden wesentliche Bestandteile und Funktionen von SimInTech beschrieben, die genutzt werden können, um die Basissoftware für eigene Erweiterungen anzupassen. Diese Funktionen wurden zur Entwicklung der Leittechnik- und Thermohydraulik-Eingabesysteme AGM und ATM für ATHLET verwendet.

5.3.2.1 Bedienoberfläche und Standardfunktionen

In der Basisversion der Software wird die Modellierung und Simulation dynamischer Systeme mit einer Vielzahl von Elementen für Anwendung in der Regelungstechnik sowie elektrische und hydraulische Netze unterstützt. Diese sind in einer Standardobjektbibliothek gespeichert und für viele Elemente ist auch eine Simulation des dynamischen Verhaltens möglich.

Nach dem Start der Bedienoberfläche kann man ein dynamisches System in einem Diagrammfenster erstellen oder modifizieren, in dem man es über das File-Menü lädt. In Abb. 5.6 ist ein Beispiel für eine typische Anwendung mit der Hauptmenüleiste von SimInTech dargestellt.

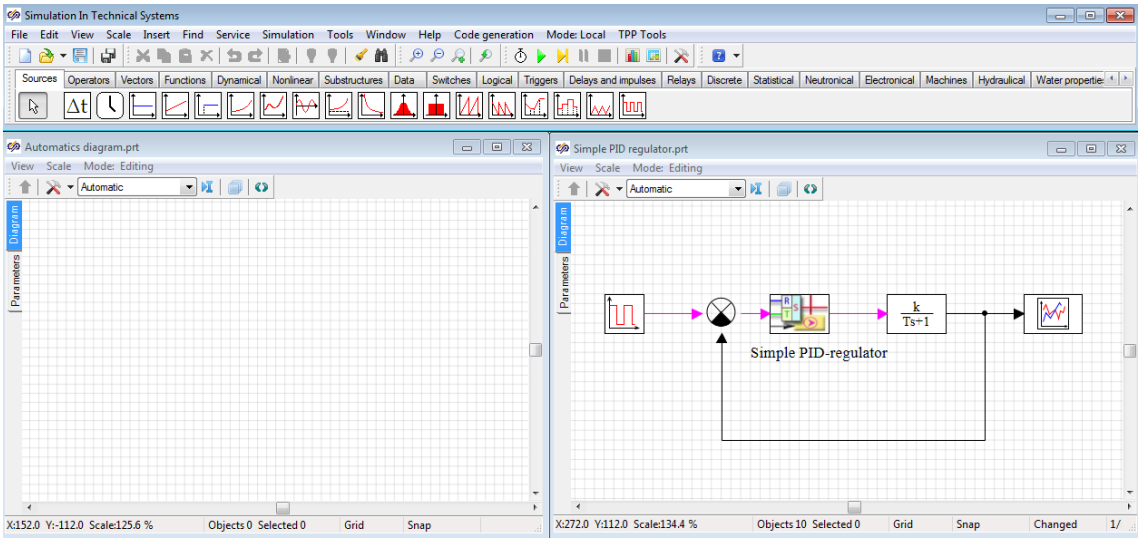


Abb. 5.6 Hauptmenüleiste und Diagrammfenster in SimInTech

Die verschiedenen Simulationsblöcke, sind entsprechend ihrer Funktionalität in unterschiedlichen Gruppen zusammengefasst. Durch „Drag and Drop“ des benötigten Elements auf die Diagrammebene und die Verbindung der Blöcke mit Linien zum Signalfluss wird das gewünschte System sukzessive erstellt. Zur Spezifikation der Elementdaten, sowohl für generelle Eigenschaften, wie Hintergrundfarbe, als auch spezifische, wie z. B. eine Zeitkonstante, werden in standardisierten Eingabedialogen verwendet (siehe Abb. 5.7). Ein System kann jederzeit in einem Binärformat oder XML-Format gespeichert werden. Eine Überprüfung einer Systemdefinition auf Vollständigkeit, wie z. B. nicht verbundene Ein- uns Ausgänge der Elemente finden erst beim Starten der Simulation des Systems statt.

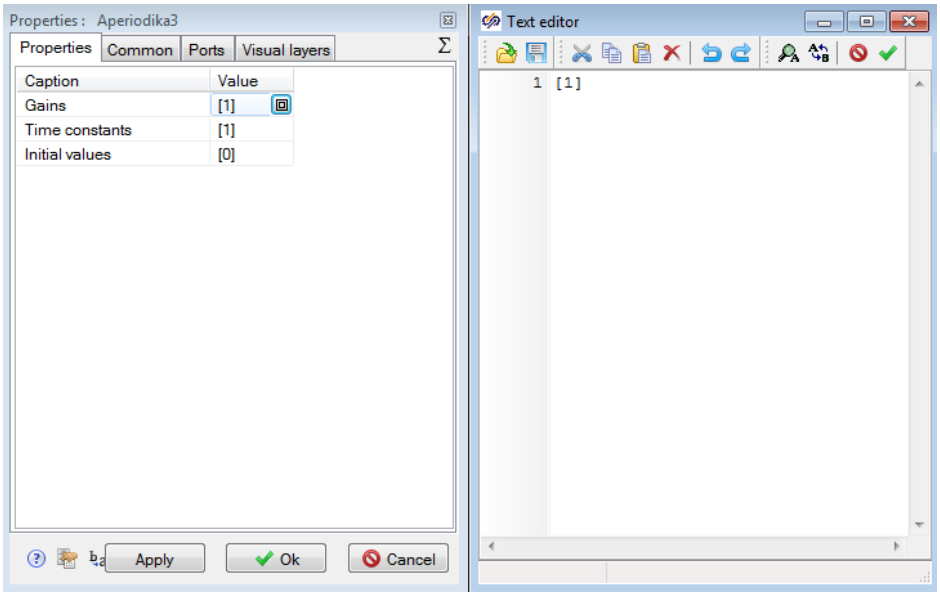


Abb. 5.7 Eingabe von Elementeigenschaften in SimInTech

Komplexe Elemente, die aus einfachen Elementen zusammengeschaltet sind, wie beispielsweise Reglerbausteine, können in sogenannten „Submodels“ zusammengefasst werden, die mit einem eigenen Symbol und unterschiedlichen Parametren mehrfach verwendet werden können. Als Beispiel ist in Abb. 5.8 ein PID-Regler gezeigt, der aus einem Proportional-, Integral und Differentialelement erzeugt wird. Die Anzahl der Anschlüsse, also Ein- und Ausgänge, die ein Submodel nach außen zur Verfügung stellt, werden ebenfalls im Diagramm für das Submodel definiert. Das Konzept von Submodels kann auch genutzt werden um sehr große, komplexe System aufzuteilen und zu strukturieren.

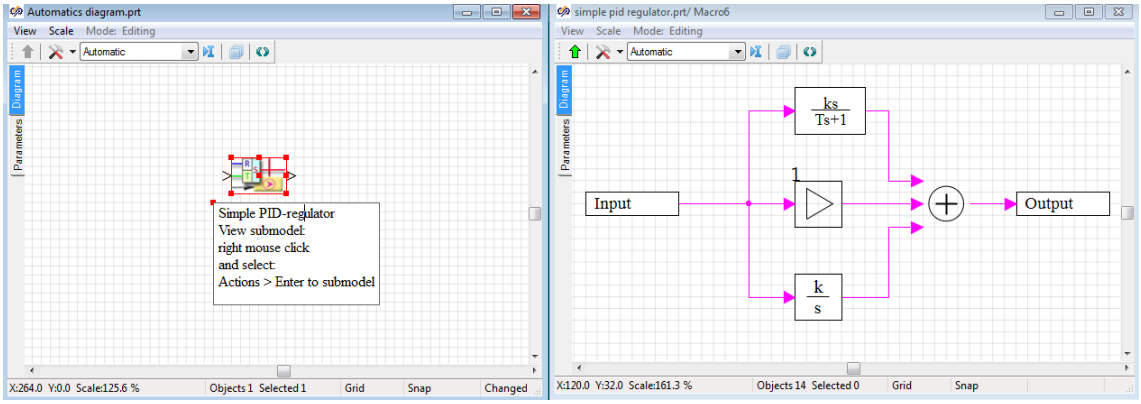


Abb. 5.8 Submodels in SimInTech

Besonders in solchen sehr großen Systemen ist es wichtig, einzelne Blöcke über ihren eindeutigen Blocknamen zu finden. In SimInTech sind dazu mehrere Verfahren möglich. Es ist möglich, wie in Abb. 5.9 gezeigt, in der Diagrammebene den „Project Tree“ anzuzeigen, der in alphabetischer Form, unter Berücksichtigung der Submodels, alle Elemente mit ihrem Namen in einer Baumdarstellung auflistet, Durch ein Doppelklicken eines Elementnamens im Baum wird das Element im Blockdiagramm angezeigt und selektiert.

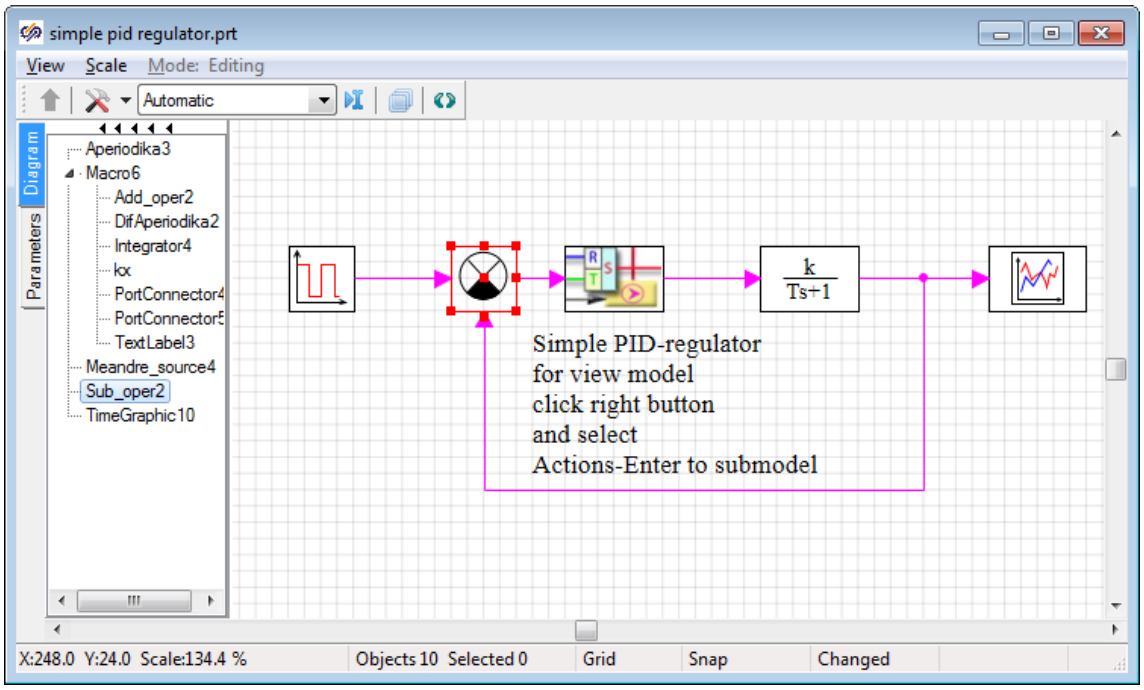


Abb. 5.9 Project Tree in SimInTech

Eine weitere Funktion zum Suchen von Elementen ist in einen Suchdialog (Abb. 5.10) implementiert, der Namen entsprechend vorgegebener Suchmuster filtert und die gefundenen Elemente auf Anforderung selektiert. Die Implementierung dieser Suchfunktion berücksichtigt allerdings gegenwärtig noch keine Elemente, die in Submodels verwendet werden. Ein ähnlicher Dialog ist für eine Suche entsprechend dem Elementtyp verfügbar.

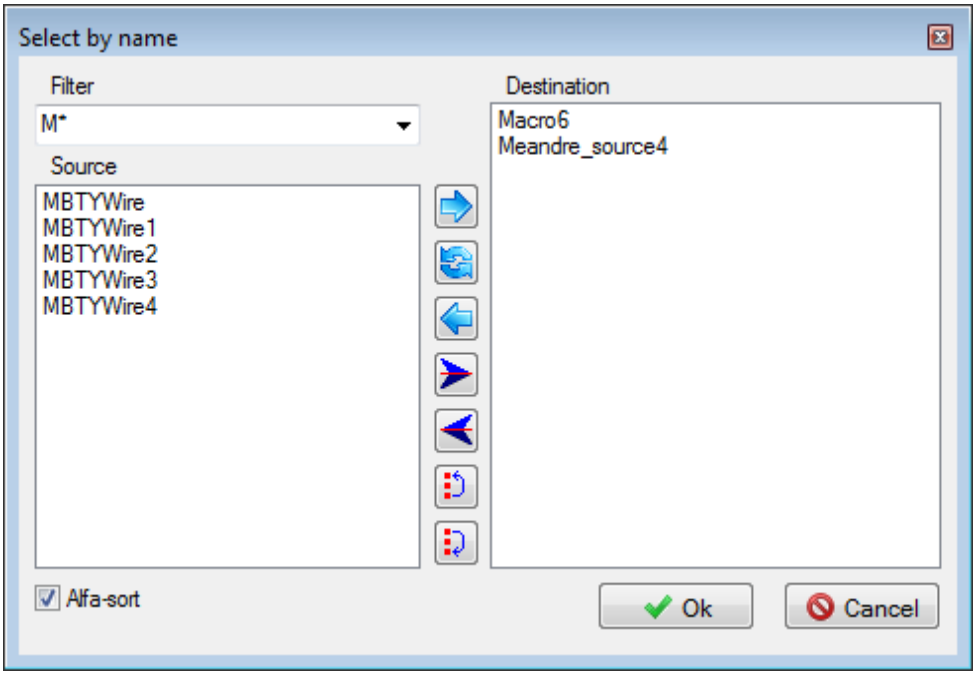


Abb. 5.10 Elementsuche in SimInTech

5.3.2.2 Ausführung einer Simulation

Nachdem alle Elemente sowie die notwendigen Daten und Signalverbindungen für ein System definiert sind, kann eine interne dynamische Simulation in SimInTech durchgeführt werden. Dazu ist Voraussetzung, dass für die verwendeten Elementtypen ein Simulationsalgorithmus definiert ist. Für viele vorhandene Standardelemente ist dies der Fall, bei benutzerdefinierten Elementen (siehe Abs. 5.3.2.3) muss dieser programmiert werden.

Vor dem Start einer Simulation können bei Bedarf wichtige Parameter zur Durchführung bei Bedarf angepasst werden, in dem der Dialog für die „Simulation Properties“ (Abb. 5.11) über das  -Symbol verwendet wird:

- Maximal step: Diskreter Simulationszeitschritt
- End time: Simulationszeit
- Show wire values: Anzeige der numerischen Ergebniswerte für alle Elemente
- Synchronize with real time: Echtzeitsimulation oder so schnell, wie möglich

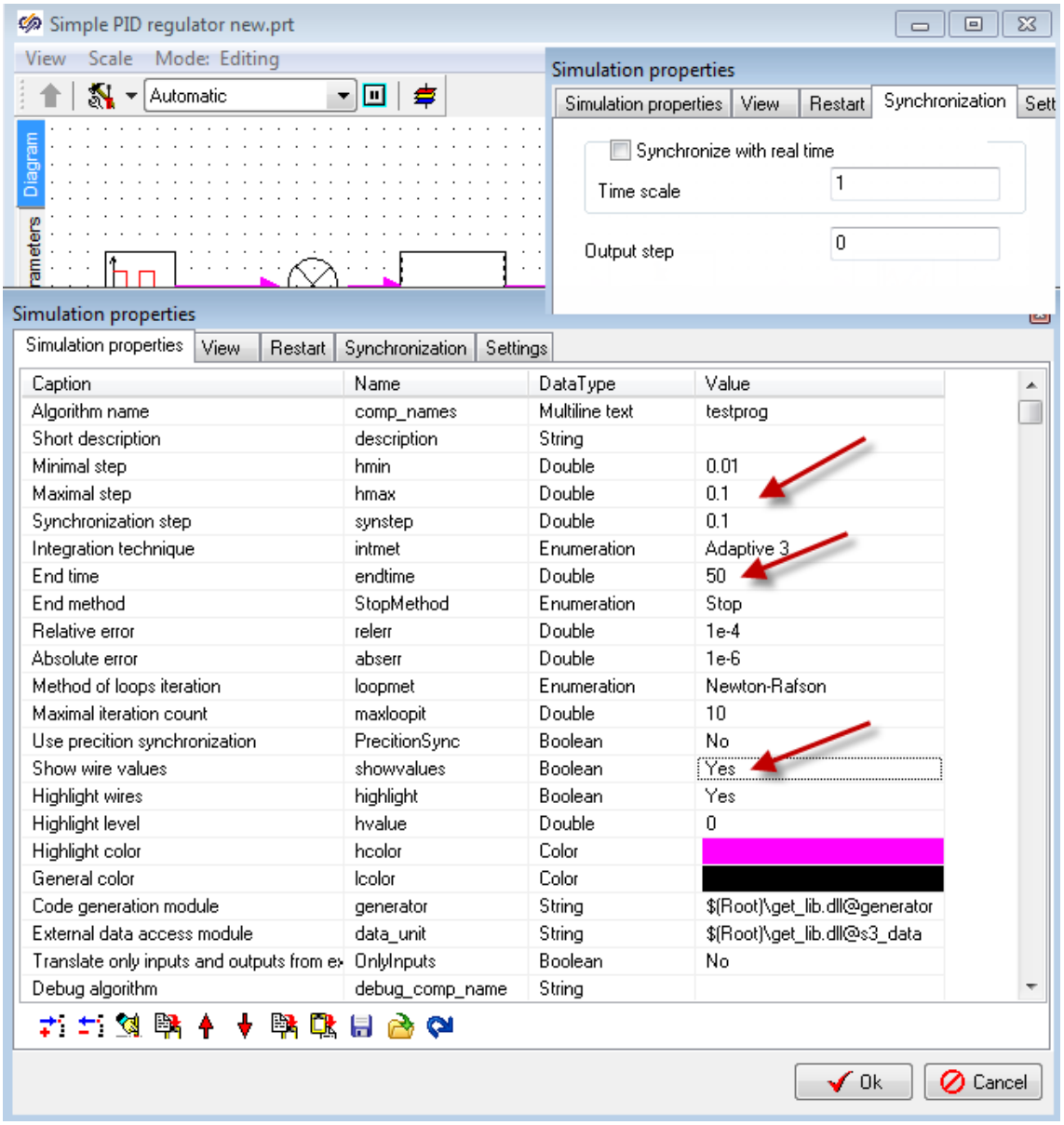


Abb. 5.11 Simulation Properties in SimInTech

Die Ausführung der Simulation, mit Start, Run, Pause und Stop, ist über den Menüeintrag „Simulation“ oder entsprechende Icons in der Hauptbedienleiste steuerbar (Abb. 5.12).

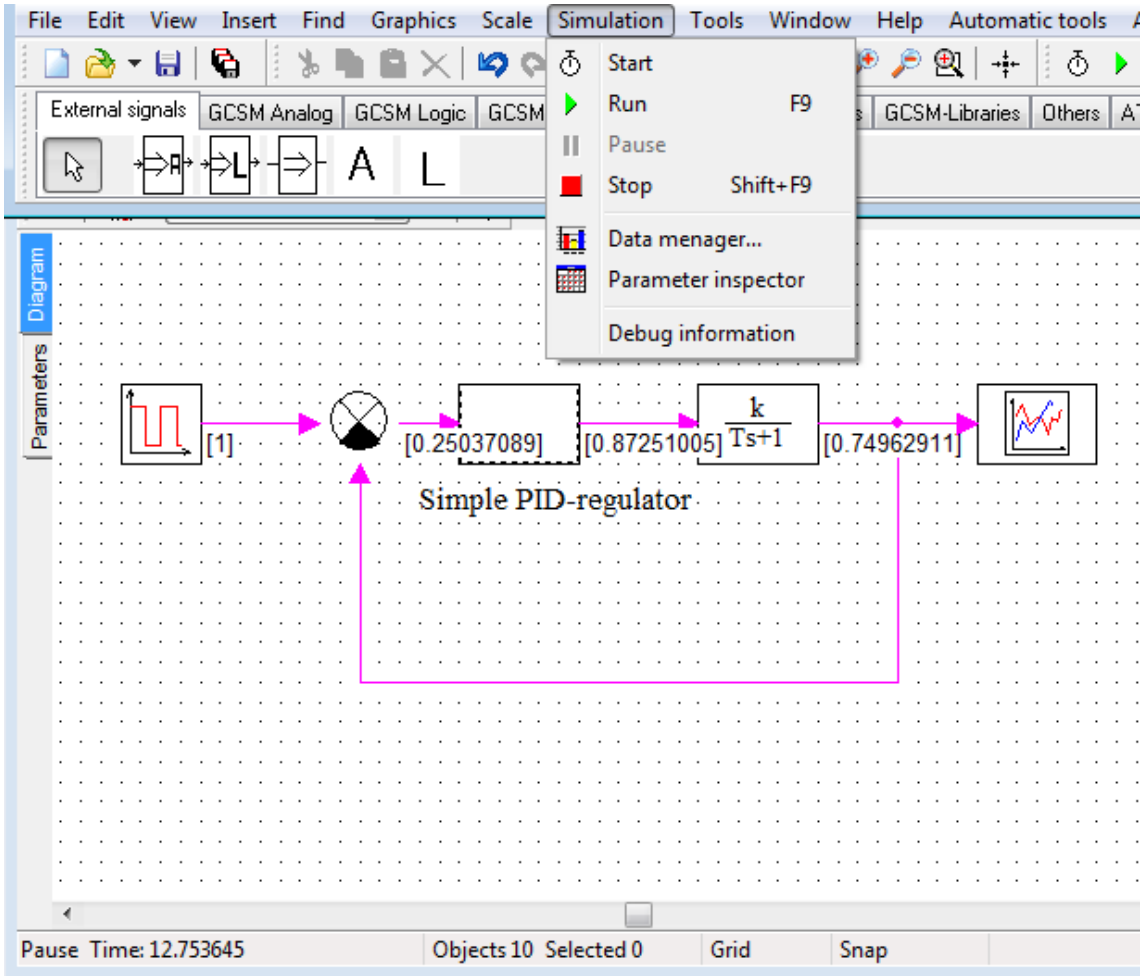
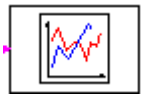


Abb. 5.12 Ausführung der Simulation in SimInTech

Unmittelbar nach dem Start der Simulation wird in SimIntech automatisch eine formale Prüfung des modellierten Systems durchgeführt. Insbesondere Prüfungen der Elementparameter und die Vollständigkeit der Verbindungen, d. h. nicht verbundene Ein- und Ausgänge der Elemente. Bei eventuellen Fehlern wird die Simulation nicht durchgeführt und die Namen der betroffenen Elemente werden in der Statuszeile des Diagramms angezeigt. Ein Mausklick auf den Namen zeigt das betroffene Element im Diagramm an.

Neben der numerischen Anzeige der Signalwerte im Diagrammfenster sind zusätzlich Darstellungen des zeitlichen Verlaufs in Achsendiagrammen möglich. SimInTech stellt dafür ein spezielles Element  für die Visualisierung bereit, das einen oder mehrere Ergebniswerte als Eingangssignale verwendet. Die Darstellung, z. B. Farben

oder Achsengitter, kann man in einem Einstellungsdialog (siehe Abb. 5.13) auf vielfältige Weise anpassen.

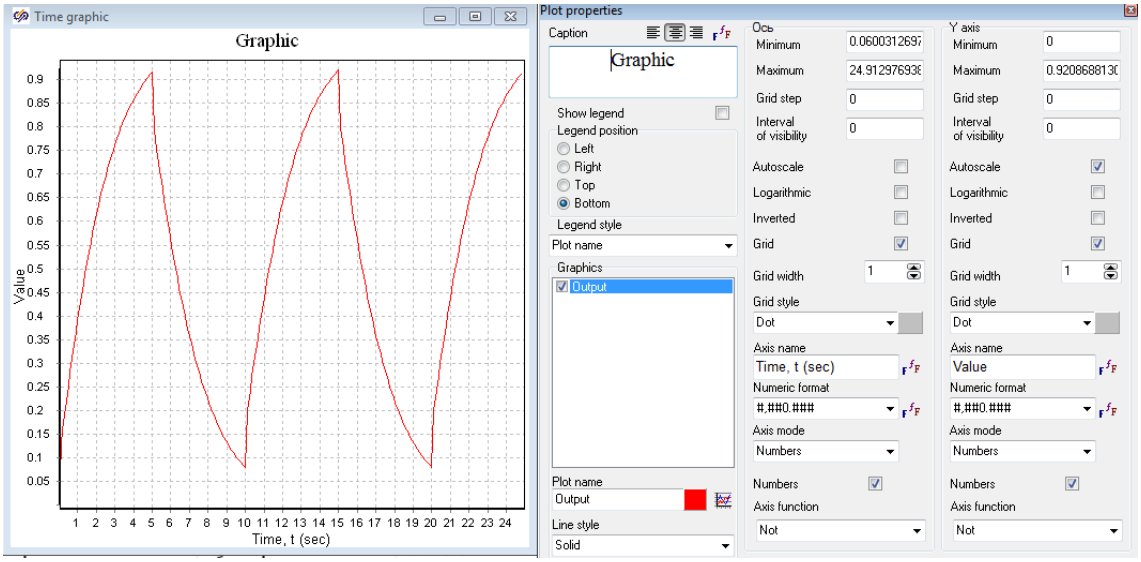


Abb. 5.13 Achsendiagramme in SimInTech

5.3.2.3 Erweiterung der Objektbibliothek

Die Erstellung von benutzerdefinierten Elementen, wie sie beispielsweise in Abs. 5.4 für die GCSM Leittechniksimulation nötig sind, erfordert mehrere Schritte, die Definition der Objektgrafiken und-parameter, die Programmierung des dynamischen Verhaltens für die Simulation und bei Bedarf eine Prozedur zur Ausgabe der Daten.

Im ersten Schritt kann ein neues Element von einem generischen oder auch einem ähnlichen, existierenden Element abgeleitet werden. Die Definition der Objekteigenschaften ist im „New blocks editor“ (Abb. 5.14) möglich. Im Reiter „Properties“ können Datenparameter definiert und geändert werden. Es stehen skalare und vektorielle Daten in verschiedenen Typen, wie Integer, Double oder String, zur Verfügung. Die Typen werden bei der Dateneingabe durch den Benutzer sofort geprüft. Die Namen der Properties können im Programm zum Zugriff auf die Daten verwendet werden.

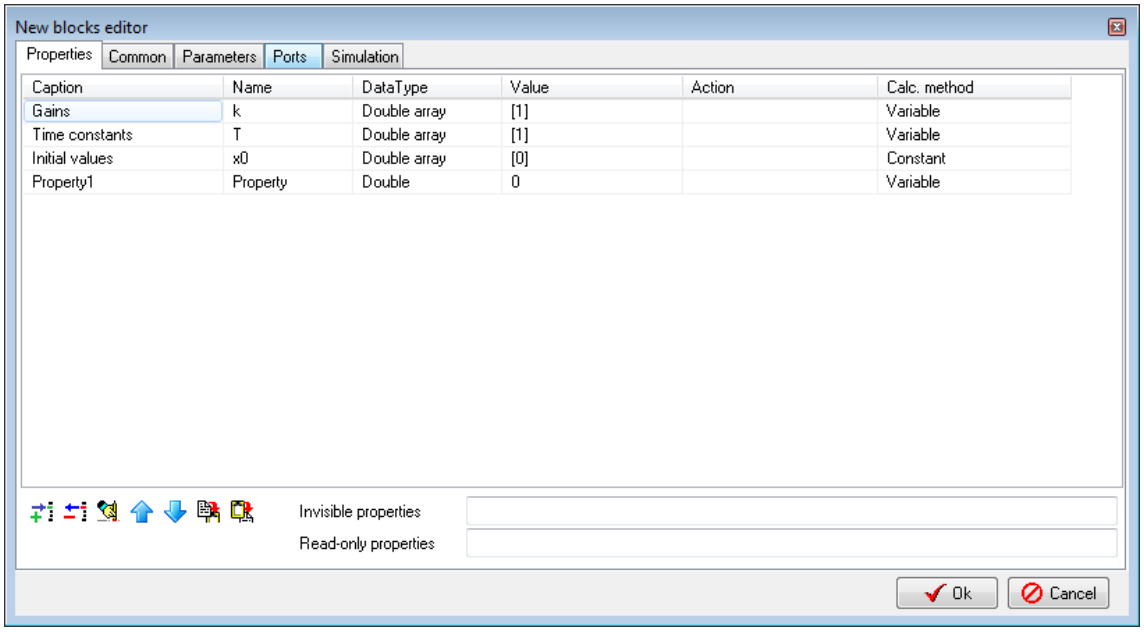


Abb. 5.14 New blocks editor in SimInTech

Für den neuen Block kann eine passende Blockgrafik erzeugt werden. Dafür steht ein Editor (Abb. 5.15) zur Verfügung mit dem die Grafik erstellt oder angepasst werden kann. Der Editor stellt eine Reihe von Basiselementen, wie z. B. Rechteck, Kreis, Linienzug, zur Verfügung. Aus diesen Elementen, die stufenlos skalierbare Vektorgrafiken sind, können auch sehr komplexe Objektgrafiken, wie z. B. Analoginstrumente, konstruiert werden.

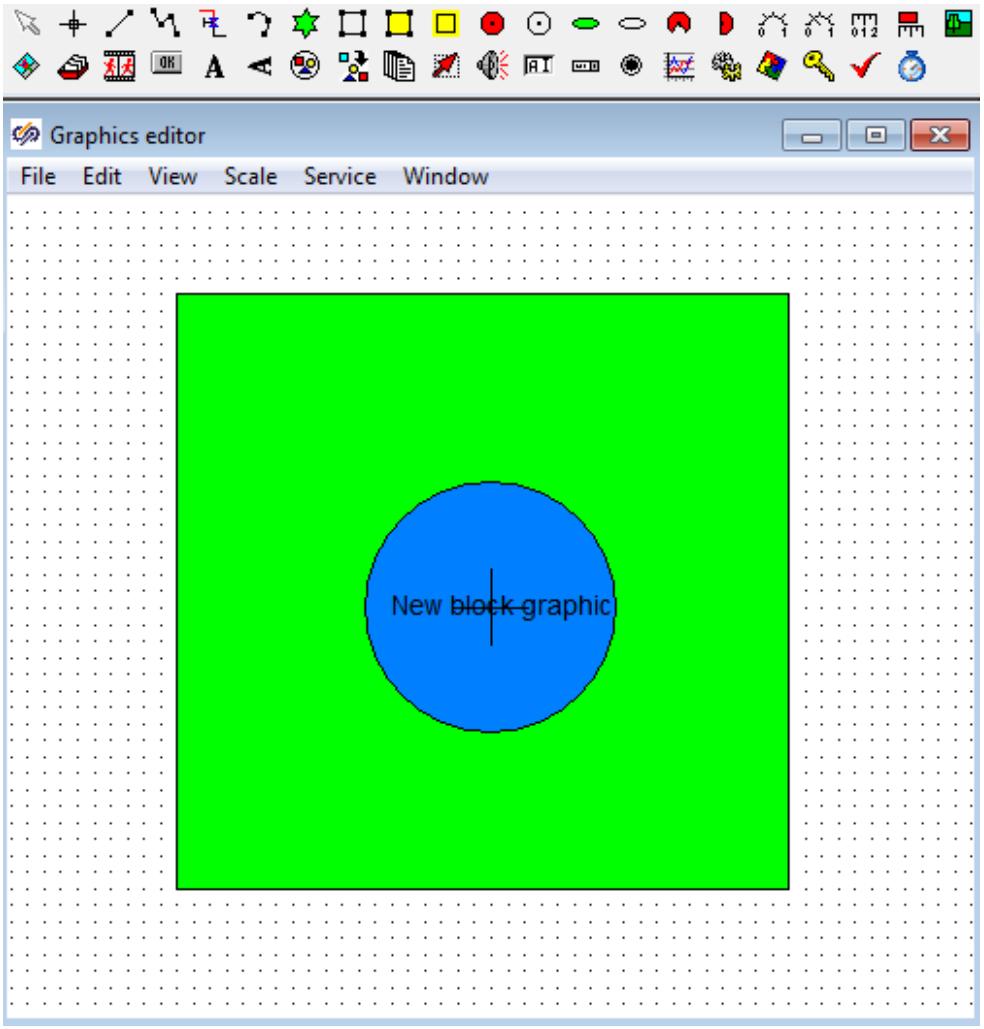


Abb. 5.15 Block graphics editor in SimInTech

Die im Bildeditor erstellte Grafik ergibt dann das folgende, skalierbare neue Blocksymbol:

Zusätzlich müssen die benötigten Ein- und Ausgänge definiert werden. Dies ist mit dem „Port Editor“ (Abb. 5.16) möglich. Dieser erlaubt es, beliebig viele Ein- und Ausgänge zum Block hinzuzufügen und die Eigenschaften, wie Typ oder Position, vorzugeben.

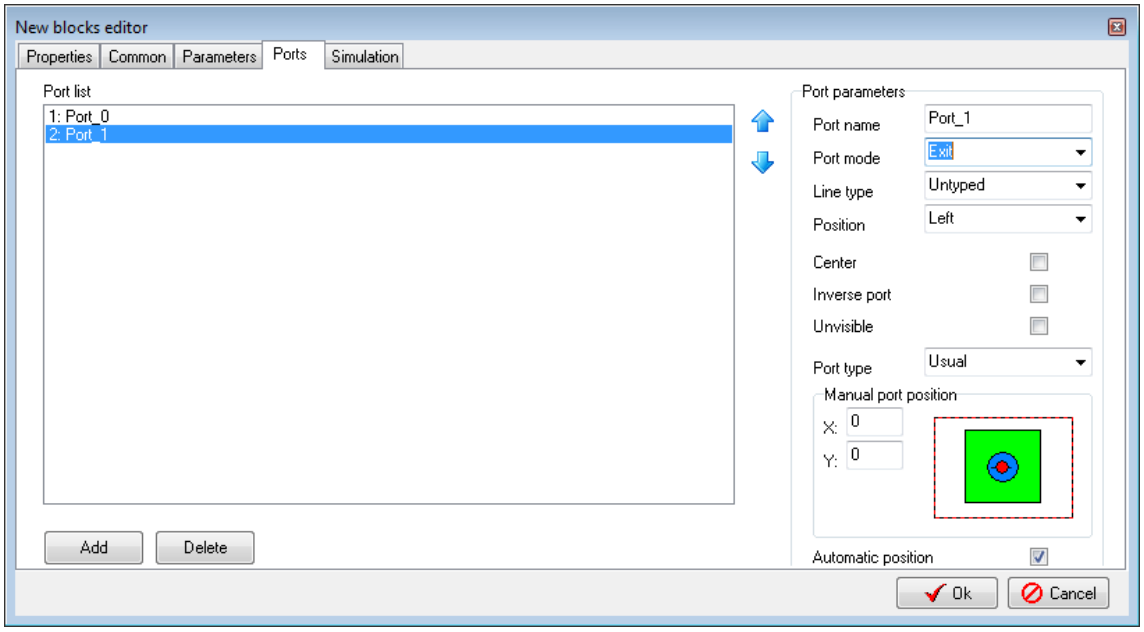
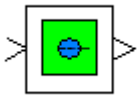


Abb. 5.16 Port Editor in SimInTech

Die im Port Editor erstellten Ein- und Ausgänge werden dann für den für den Block beispielsweise folgendermaßen berücksichtigt:



Zuletzt kann man noch festlegen, wie das dynamische Verhalten des Blocks simuliert werden soll. Dies ist entweder durch die Nutzung der internen Programmiersprache von SimInTech möglich oder durch die Bereitstellung einer externen Simulationsbibliothek.

Für interne Programme gibt es einen speziellen „Programming Language Block“, der der Programmcode zur Bestimmung des Berechnung von den Ausgängen aus den Eingängen enthält. Wenn man einen Block dieser Art seinem eigenen Block als „Submodel“ zuordnet, wird der Programmcode zur Simulationszeit mit den aktuellen Eingangswerten ausgeführt. Eine Dokumentation der internen Programmiersprache ist im „SimInTech User Manual“ /SIT 15B/ enthalten.

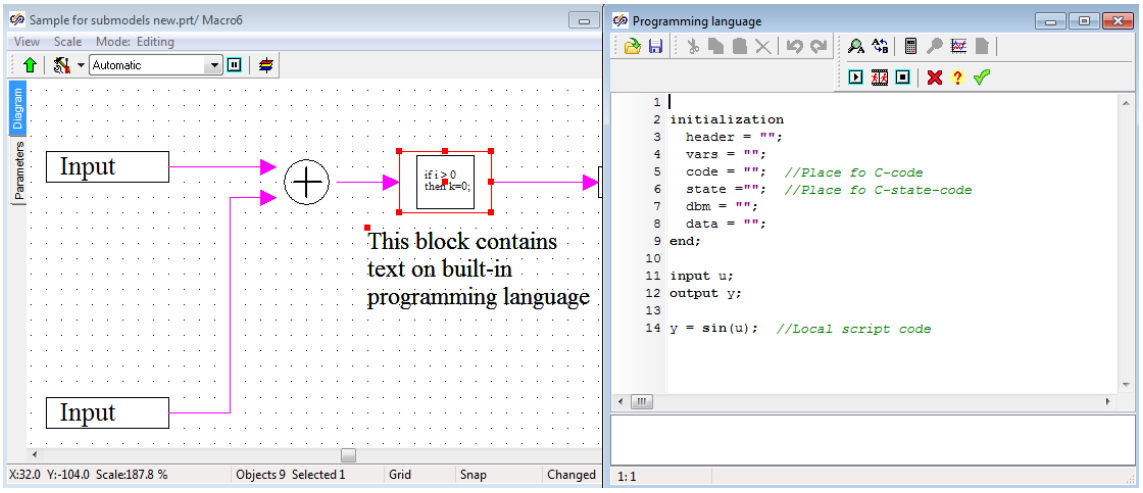


Abb. 5.17 Interne Programmblöcke in SimInTech

Bei der Einbindung einer externen Simulationsbibliothek muss in der Beschreibung des Blocks im Reiter „Simulation“ der Name der Bibliothek (Dll) und ein Verweis auf den Simulationscode (Abb. 5.18) eingetragen werden.

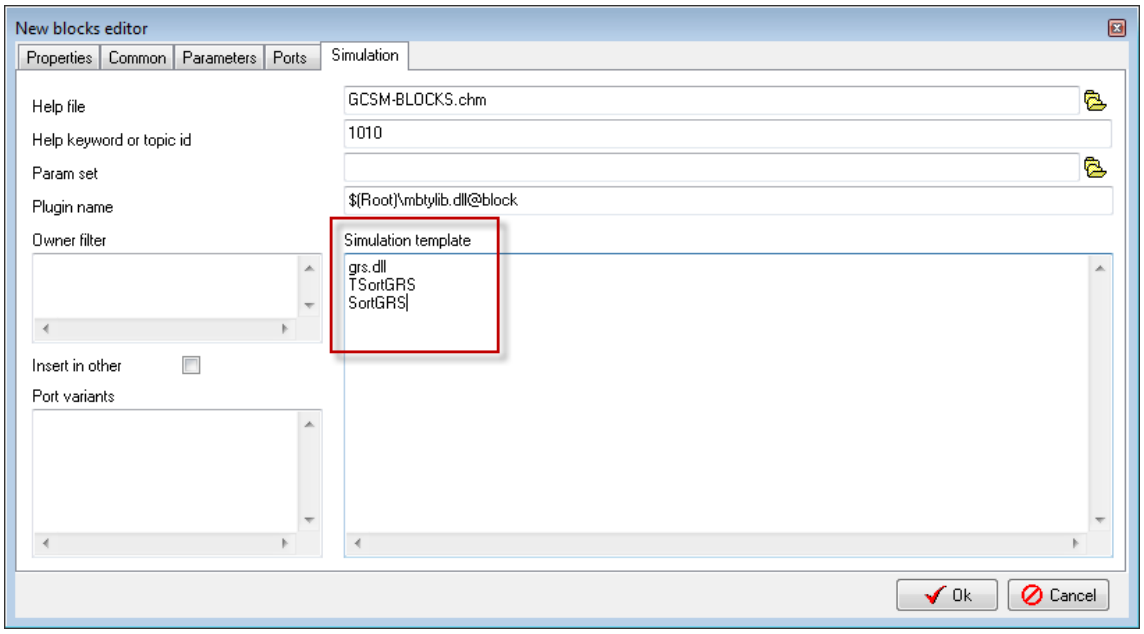


Abb. 5.18 Externe Programmbibliothek in SimInTech

Im letzten Schritt muss das neue Objekt dann permanent gespeichert werden. Dazu sollte ein neuer Klassenname (Abb. 5.19) für das Objekt vergeben werden. Anschließend kann das neue Objekt in der Klassenbibliothek (Menüpunkt „File > Save to library“) gespeichert werden.

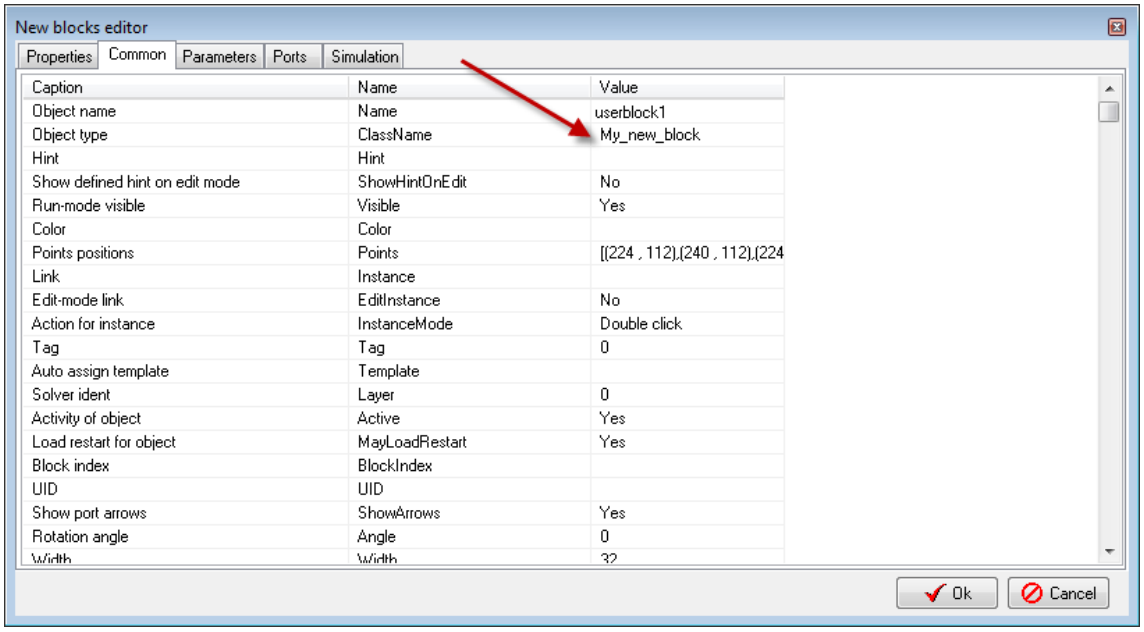


Abb. 5.19 Neue Objektklasse in SimInTech

Es ist nun noch erforderlich, den neuen Objekttyp in der Symbolleiste für die verfügbaren Blöcke hinzuzufügen, damit der Benutzer den neuen Block, wie alle anderen, auf die Diagrammebene platzieren kann. Dazu kann mit dem Klasseneditor (Menüpunkt „File > Edit class library“) ein Symbol für den neuen Block in einem beliebigen Reiter der Hauptbedienleiste (Abb. 5.6) einfügt werden.

5.3.2.4 Erweiterung in der Entwicklungsumgebung

Wie aus den vorhergehenden. Abschnitt erkennbar ist, können bereits einige Anpassungen ohne weitere Änderungen des SimInTech Quellcodes durchgeführt werden. Benötigt man umfangreichere Erweiterungen, beispielsweise neue Dialoge oder Zusatzfunktionen wie die Datenausgabe für ATHLET, können die Änderungen in der Entwicklungsumgebung „Delphi XE“ /EMB 15/ durchgeführt werden. Delphi ist eine kommerzielle Entwicklungsumgebung (Abb. 5.20) für die Programmiersprache „Object Pascal“ und bietet einen hohen Funktionsumfang, wie Klassenbrowser, kontextsensitiven Programmeditor und umfangreiche Debugoptionen.

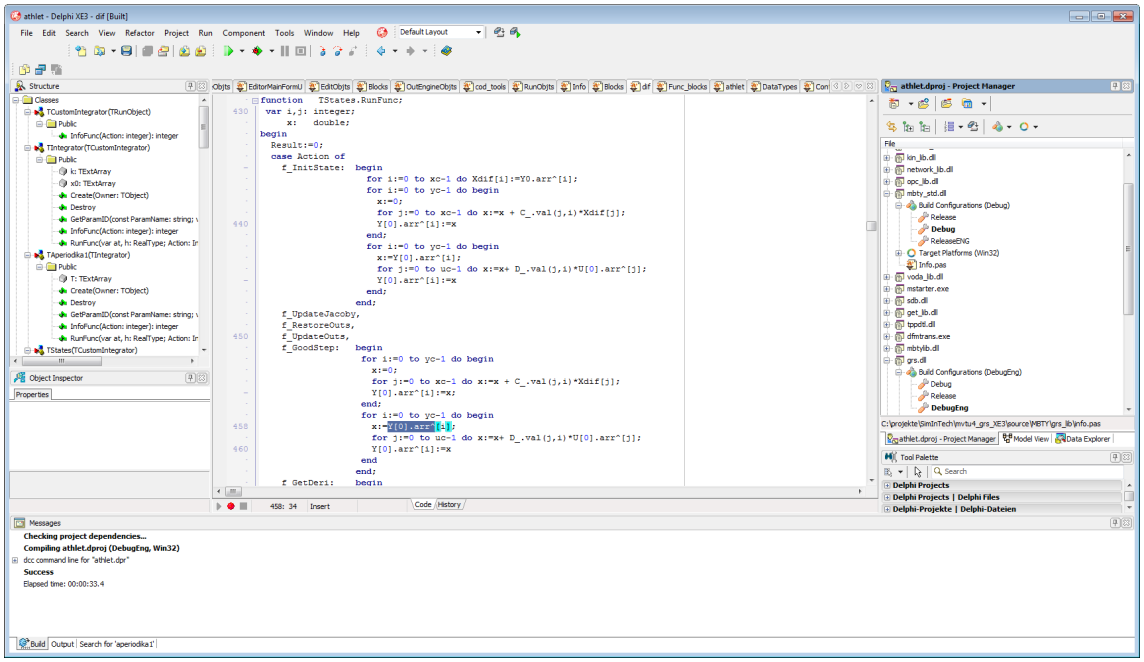


Abb. 5.20 Entwicklungsumgebung „Delphi“ für SimInTech

Es ist vorteilhaft, die Erweiterungen möglichst in einer eigenen Laufzeitbibliothek durchzuführen und die einzelnen Funktionen über standardisierte Prozedurschnittstellen an den „Kernel“ von SimInTech zu übergeben. Dieses Vorgehen erleichtert es, Verbesserungen in der Standardversion des Herstellers auf einfache Weise, im Idealfall durch ein „Update“ im Versionsmanagementsystem SVN zu übernehmen. Das Name der eigenen Laufzeitbibliothek wird beim Starten von SimInTech in den Programmeinstellungen eingetragen (Menüpunkt „File > Parameters > Plugins“)und das Laden den Bibliothek erfolgt dann in der Initialisierungsroutine „LoadExtension-Library“. Die Basissoftware ist grundsätzlich sehr objektorientiert aufgebaut und gut modularisiert. Die Kommentierung im Programmcode ist allerdings überwiegend in russischer Sprache vorhanden. Programmerweiterungen können aber bei Bedarf auch mit der Unterstützung des Herstellers erfolgen. Dieses Vorgehen setzt einen entsprechenden Supportvertag voraus. Damit ist dann sichergestellt, dass die Erweiterungen möglichst optimal implementiert werden können und sich der Aufwand, z. B. bei der Fehlersuche, verringert.

5.4 Adaption und Anwendung des Netzwerkeditors für die Leittechnikmodellierung in ATHLET

Das Programm SimInTech wurde so erweitert, dass die GCSM-Signalelemente aus dem aus dem bisher verwendeten GCSM-Generator und dessen Funktionalität nun auch in dem neuen ATHLET GCSM Modeller (AGM) zur Verfügung stehen. Zur Simulation dieser Signalelemente wurden Programme entwickelt bzw. aus dem GCSM-Modell von ATHLET übernommen und in SimInTech integriert. Eine neue Funktion erlaubt es zu einem, mit Hilfe des Netzwerkeditors erstellten Diagramm, die äquivalenten ASCII-Dateien zu erzeugen, die direkt in die ATHLET- Eingabe für das GCSM-Modell eingebunden werden können. Mit Hilfe einer weiteren Funktion kann eine Datei im APG-Format zur Anzeige in ATLAS erzeugt werden. Die Entwicklung von AGM hat einen Stand erreicht, der die praktische Anwendung, auch für umfangreiche Leittechniksysteme, ermöglicht. Zur Bedienung von AGM sind eine interaktive Online-Hilfe und eine Benutzerhandbuch /DEI 14/ verfügbar. Detaillierte Hinweise zur Programmierung sind in der Programmdokumentation /DEI 15/ zu finden.

5.4.1 GCSM-Signalelemente

Mit Hilfe der Bedienoberfläche von SimInTech wurde eine Elementbibliothek erstellt, die in sechs Gruppen geordnet die GCSM-Signalelemente enthält, die zur Erstellung eines GCSM-Diagramms benötigt werden (Abb. 5.21). Eine zusätzliche Gruppe enthält die GCSM-Subworkspaces, in der einige häufig vorkommende Elementkombinationen zu Funktionsbausteinen zusammengefasst wurden. Funktionalität und Layout der Signalelemente wurden weitgehend vom GCSM Generator übernommen. Einige Signalelemente, z. B. standardisierte Ein- und Ausgabesignale und ein Element zur grafischen Darstellung von Zeitdiagrammen, konnten von SimInTech übernommen werden.

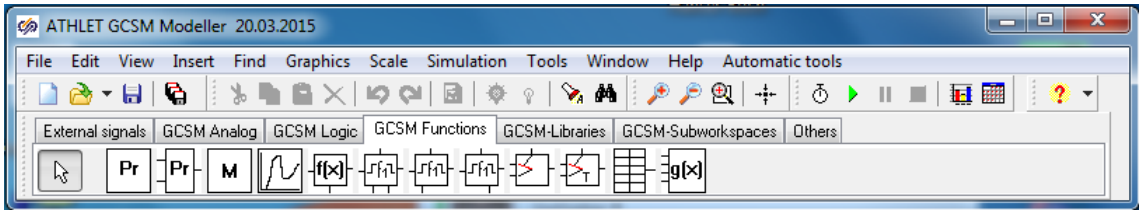


Abb. 5.21 Menü zur Auswahl der GCSM-Signalelemente

Ein GCSM-Signalelement besitzt maximal 4 Eingänge und genau einen Ausgang. Bei den meisten Signalelementen ist die Anzahl der Eingänge fest vorgegeben, bei manchen (ADDER, AND, OR, SORT, GUSER, LIBBOR) kann die Anzahl der Eingänge auch vom Benutzer spezifiziert werden. Ändert der Anwender die Anzahl der Eingangssignale, dann wird die Größe des Signalblocks an die Anzahl der Eingangssignale angepasst. Ein- und Ausgänge können logische Werte (EIN/AUS) oder arithmetische Werte (reelle Zahlen) annehmen. Jedem Signalelement ist eine Reihe von Eigenschaften zugeordnet, (Sollwert, Anfangswert des Ausgangssignals u.a.m.), die vom Anwender modifiziert werden können (Abb. 5.22). Manche Signalelemente, z. B. diejenigen, die zur Simulation ihren Ausgangswert zu einem vorhergehenden Zeitpunkt benötigen, enthalten auch Parameter, die, für den Anwender unsichtbar, im Verlauf der Simulation belegt werden.

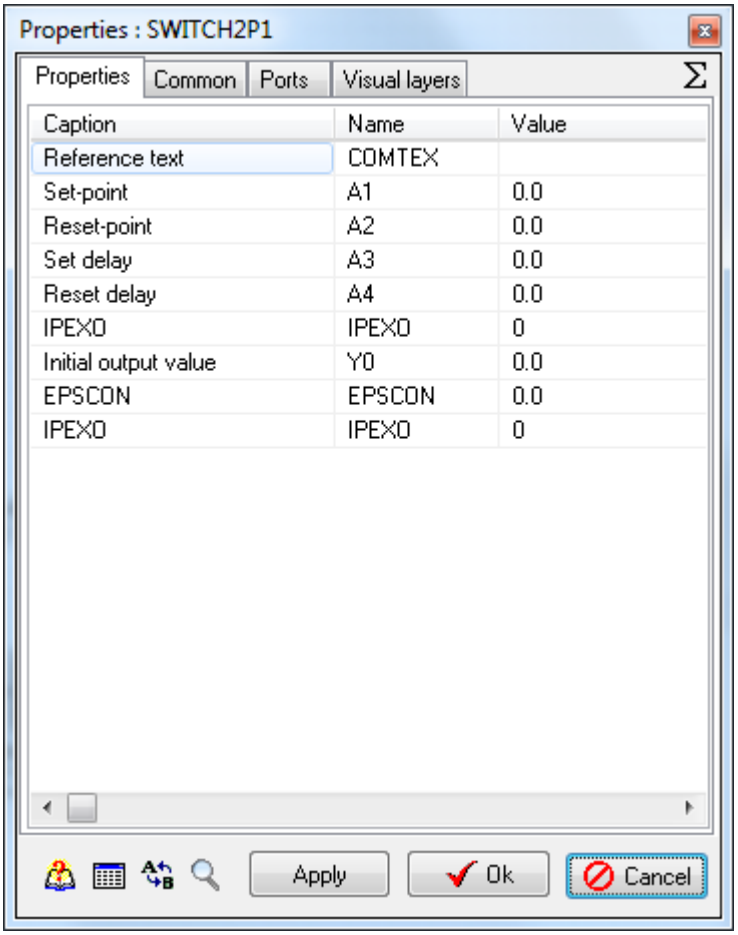


Abb. 5.22 Eigenschaften des GCSM-Signalelements SWITCH2P

Ändert ein Anwender den Wert eines Parameters, dann wird vom Programm geprüft, ob der neue Wert für diesen Parameter gültig ist. Gegebenenfalls wird eine Fehlermeldung am unteren Rand des Diagramms ausgegeben.

Viele Signalelemente enthalten einen Parameter, mit dem der Wert der Eingangssignale multipliziert wird. Dieser Faktor wird im Allgemeinen innerhalb des Blocks rechts neben dem Eingangssignal angezeigt (Abb. 5.23).

Für die meisten Ausgangswerte ist ein Startwert Y0 vorgegeben. Ist dieser von 0 verschieden, dann wird er im Allgemeinen rechts neben dem Signalblock über dem Ausgangssignal angezeigt

Auch der Faktor, mit dem der errechnete Wert des Ausgangssignals multipliziert wurde, wird bei einigen Blöcken oberhalb des Signalblocks angezeigt.

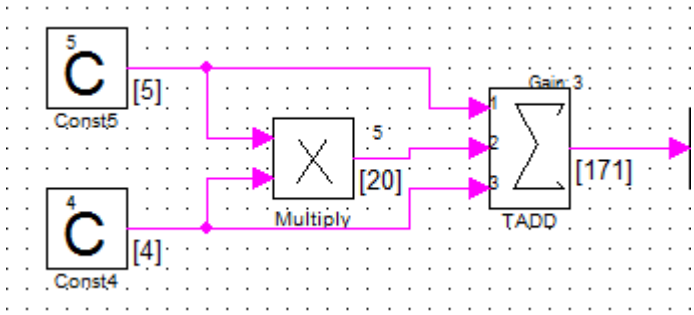


Abb. 5.23 Anzeige von Parametern im GCSM-Diagramm

5.4.2 Erstellung von ATHLET Eingabedaten für die GCSM Signalelemente

Wird vom AGM Hauptmenü unter dem Reiter Tools der Abschnitt „Create ATHLET GCSM Input“ (Abb. 5.24) aktiviert, dann wird folgende Eingabemaske angezeigt:

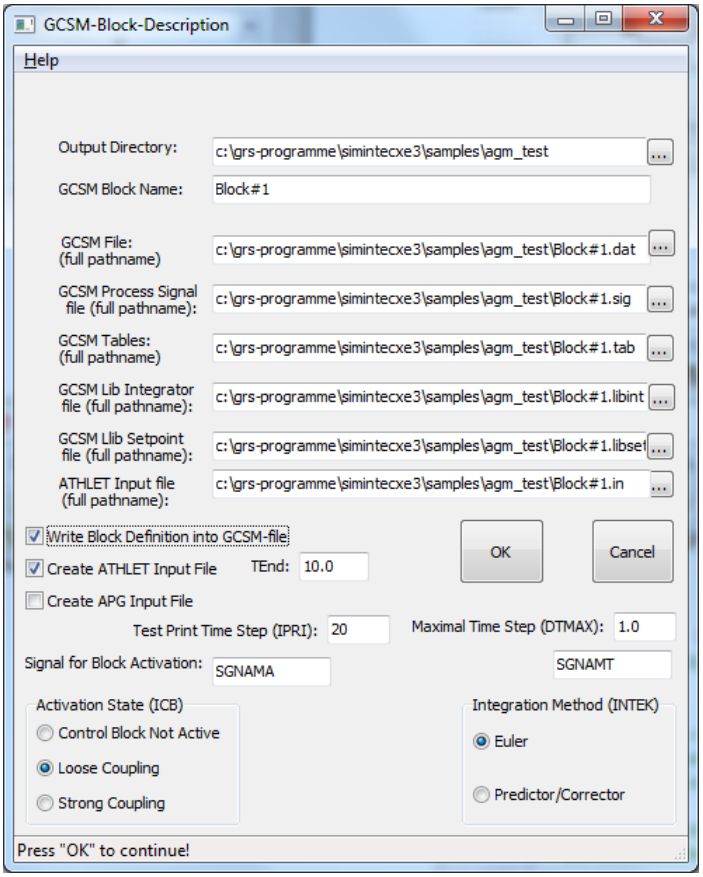


Abb. 5.24 Eingabemaske zur Erstellung von ATHLET-GCSM Eingabedaten

Hier werden die Namen der Dateien spezifiziert, die die Beschreibung der GCSM-Signalelemente enthalten. Zusätzlich zu diesen Dateien, die über geeignete Include-Anweisungen in einen bereits vorhandenen ATHLET Eingabedatensatz übernommen werden können, kann auch ein vollständiger ATHLET Eingabedatensatz zum Testen des GCSM Modells erzeugt werden.

Bei der Aktivierung von ‚Create APG Input File' wird das GCSM-Diagramm im APG-Format gespeichert und kann dann als Eingabedatei zur Erzeugung des Systemdiagramms für ATLAS verwendet werden.

5.4.3 Simulation und Anwendung

In AGM kann das dynamische Verhalten der modellierten Systeme direkt simuliert und untersucht werden. Mit der Nutzung von Testsignalen für die Randbedingungen, z. B. Sprungfunktion, Rampe, Sinus, kann das Systemverhalten bereits ohne einen ATHLET Rechenlauf überprüft und ggf. angepasst werden. Zur Simulation des erstellten Diagramms, d. h. zur Berechnung der Ausgangssignale der GCSM Blöcke in Abhängigkeit von deren Eingangssignalen und den Block-spezifischen Parametern, werden dieselben Algorithmen wie in ATHLET verwendet. Einige Unterprogramme, insbesondere die zur Berechnung besonders komplexer Algorithmen, konnten nach geringen Modifikationen direkt von ATHLET übernommen werden. Dieses Vorgehen stellt eine größtmögliche Übereinstimmung der Simulationsergebnisse sicher.

Im Folgenden wird die Anwendung von AGM zur Simulation an einem Beispiel aus der Regelungstechnik /MEY 15/, einer 2-Punkt-Regelung mit Rückführung, die beispielsweise für Temperaturregelungen eingesetzt wird, näher erläutert. In Abb. 5.25 ist im linken Teil der Skizze der Regler mit Rückführung dargestellt, der für eine Regelstrecke mit Totzeitverhalten und Verzögerung 1.Ordnung verwendet wird.

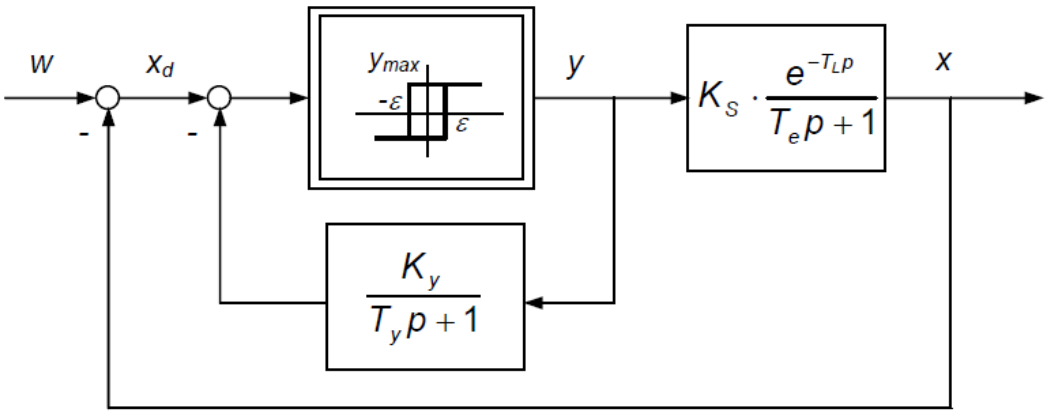


Abb. 5.25 Prinzipschaltbild einer 2-Punkt-Regelung mit Rückführung

Für ein konkretes Anwendungsbeispiel wurde diese Regelung in AGM modelliert. Die Regelstrecke wurde ohne Totzeit mit einer Zeitkonstante von 30 s modelliert. Geeignete Reglerparameter für Hysterese und Rückführung wurden im Anschluss ermittelt. Im linken Teil des Blockschaltbilds (Abb. 5.26) wird zum Test des Systems ein Sollwertsprung (0,8 auf 0,6) nach einer Zeit von 100 s angeregt. In der Simulation ist dieser mit der grünen Kurve dargestellt. Die Regelgröße, d. h. das Ausgangssignal des Elements SYSLAG ist in der roten Kurve, erkennbar.

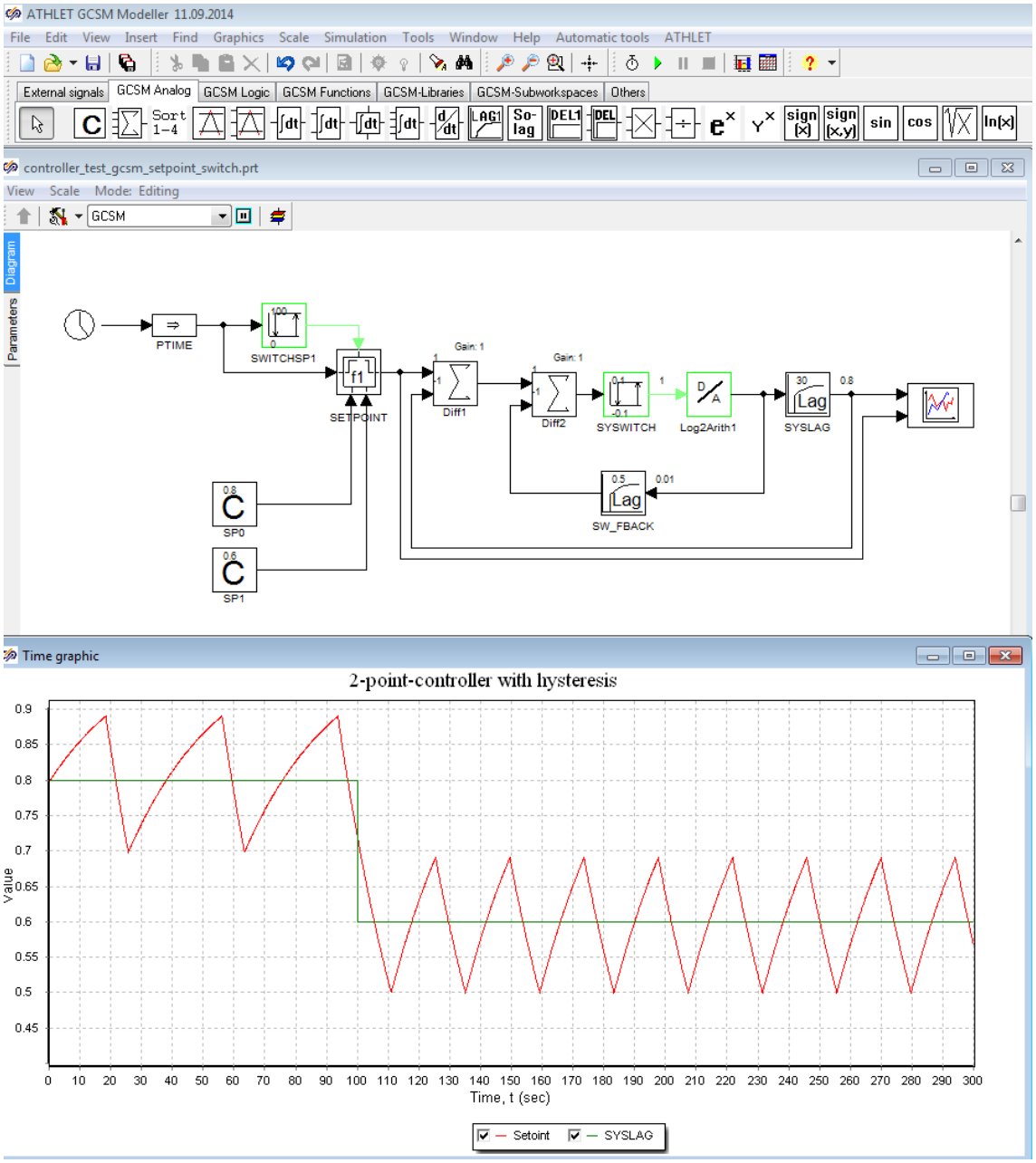


Abb. 5.26 Modellierung und Simulation der 2-Punkt-Regelung in AGM

Nach dem Export des Systems für ATHLET/GCSM und des Systemdiagramms mit ATLAS können mit den automatisch erzeugten Daten direkt eine Simulation mit ATHLET durchgeführt werden und die Ergebnisse in ATLAS dargestellt werden. In Abb. 5.27 kann man oben die automatisch erzeugte Darstellung des Systems sehen. Über den Signalelementen sind die dynamischen Ergebniswerte der Simulation eingeblendet. Im unteren Teil des Bilds ist das wieder der Zeitverlauf des Signals SYSLAG sichtbar. Es ist gut erkenn bar, das die Simulation in ATHLET für diesen Fall identische Ergebnisse wie in AGM liefert. Bei sehr komplexen Systemen können sich, beispiel-

weise aus numerischen Gründen, im Einzelfall auch Unterschiede ergeben, die dann evtl. eine Anpassung der Modellierung in AGM erfordern können.

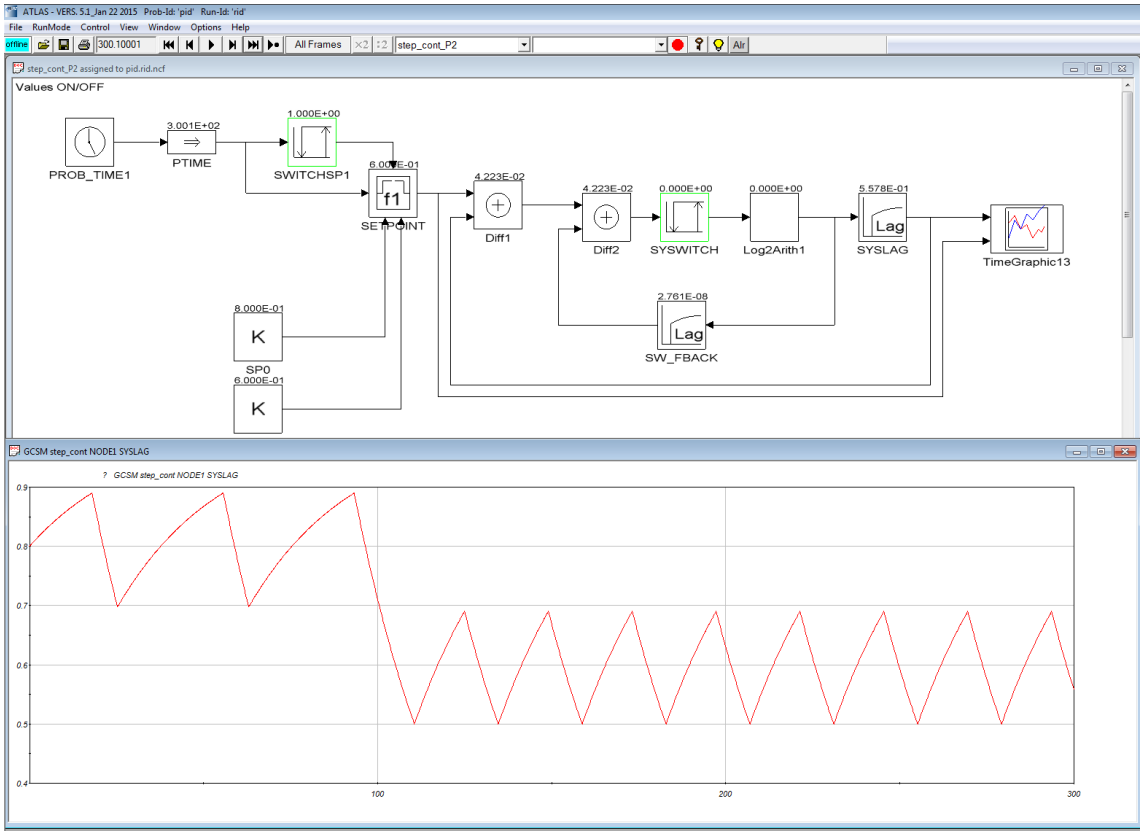


Abb. 5.27 Simulation und Visualisierung der 2-Punkt-Regelung in ATHLET und ATLAS

5.4.4 Hilfesystem zur Leittechnikmodellierung mit AGM

Für die Leittechnikmodellierung in ATHLET wurde ein umfangreiches Hilfesystem aufgebaut, das den Anwender bei jedem Schritt zum Entwurf eines Diagramms unterstützt. Insbesondere gibt es zu jedem Signalelement eine Funktionsbeschreibung, eine Beschreibung der Eingangssignale und eine Beschreibung der Parameter (Abb. 5.28), die für die jeweiligen Elemente kontextsensitiv angezeigt werden kann.

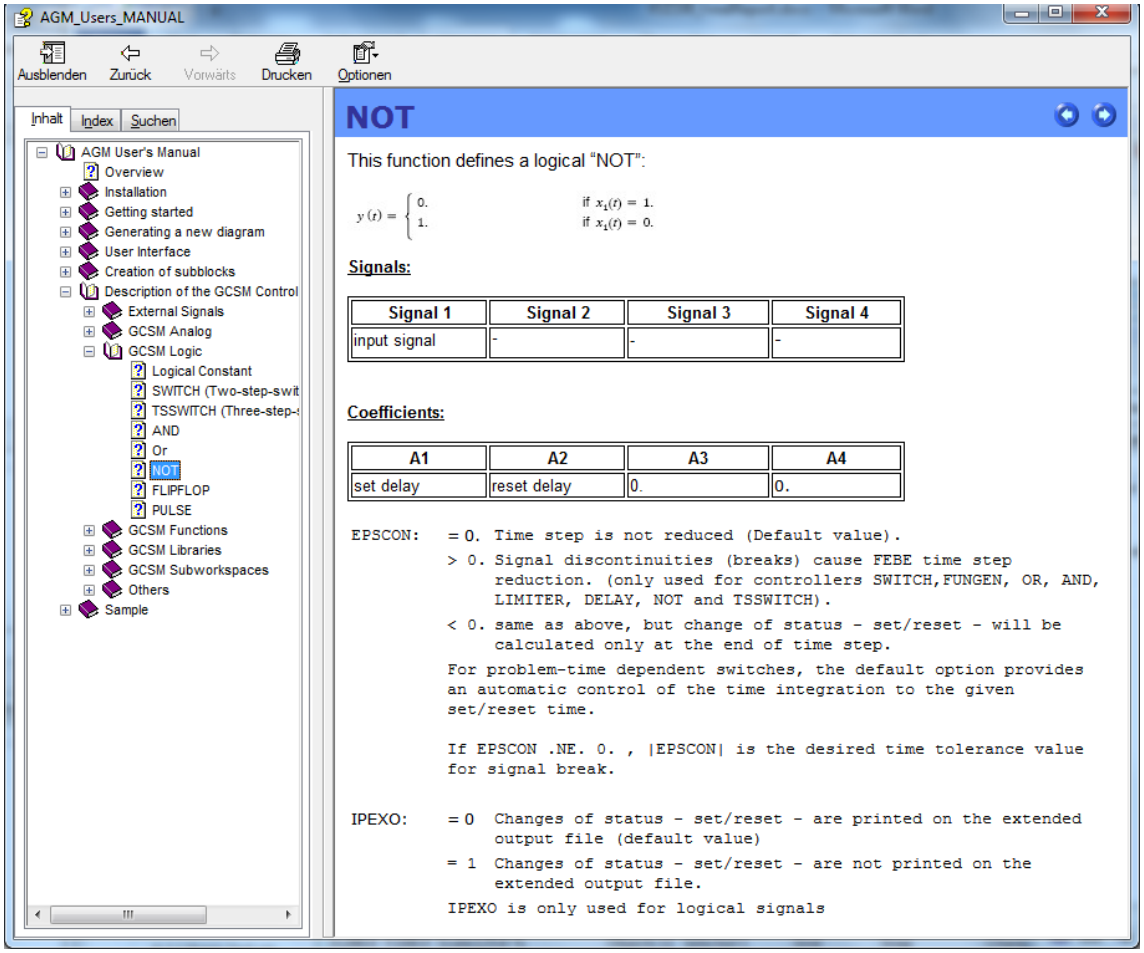


Abb. 5.28 Hilfe für das GCSM-Signalelement NOT

5.5 Erweiterung des Netzwerkeditors für die Thermohydraulikmodellierung in ATHLET

Nach den positiven Erfahrungen mit dem Einsatz von SimInTech zur Entwicklung des GCSM Eingabesystems wurde die Eignung Basissoftware für die Thermohydraulikmodellierung in ATHLET mit der Entwicklung eines Prototyps überprüft. Dieser Prototyp, der „ATHLET Thermohydraulic Modeller (ATM)“ sollte die notwendigen Funktionen und Eingabedialoge der Daten für ATHLET Thermofluid-Objekte (TFO), die Verknüpfung dieser Objekte in einem Fluidnetzwerk und die Ausgabe dieser Daten im von ATHLET lesbaren Format bereitstellen.

Zur Modellierung eines Fluidobjekts im Netzwerkeditor werden als wesentliche Funktionen benötigt:

- Eingabe der Geometriedaten durch Geometriebausteine (z. B. Düse, Rohr, Abzweigung)
- Darstellung der vorgegebenen Geometrie in 2D
- Vorgabe der ortsabhängigen Metadaten (z. B. Reibungsbeiwerte, Temperaturverteilung)
- ASCII-Ausgabemodul für das ATHLET Format

Zur Realisierung wurden zunächst entsprechende Objekte (Abb. 5.29) für die benötigten Geometriebausteine in SimInTech definiert. Entsprechend den Anforderungen aus den Isometrieplänen gibt es Rohrstücke und –bögen („Krümmer“), Abzweige sowie Pumpen und Ventile. Zur Berücksichtigung der absoluten Lage im Raum kann ein 3D-Punktobjekt verwendet werden.

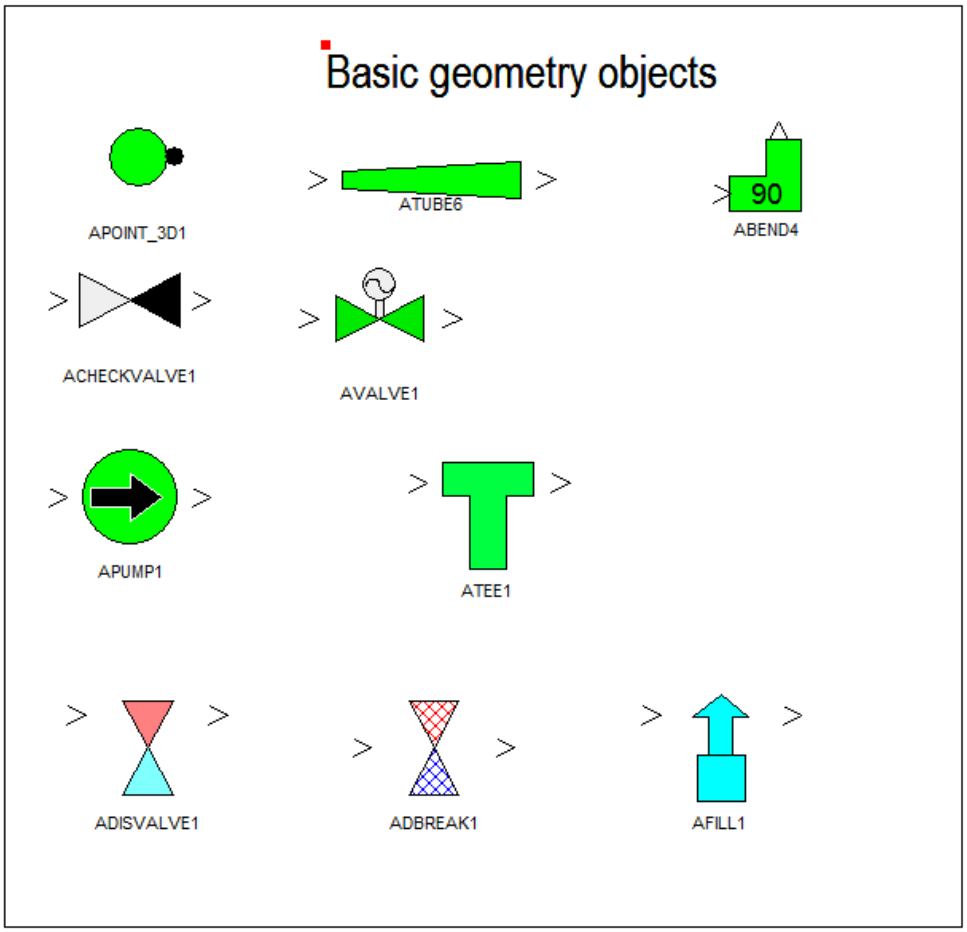
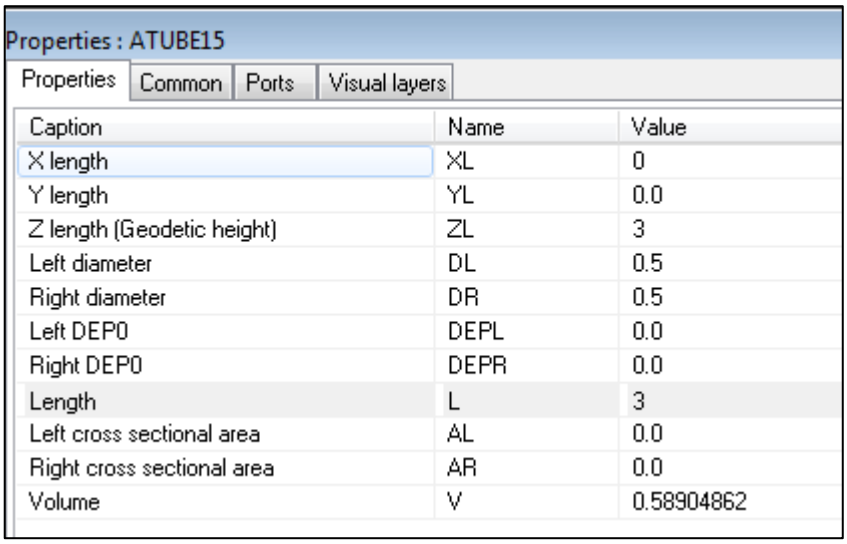


Abb. 5.29 Objekte für die Geometrie-Erzeugung in ATM

Die Objekte besitzen Daten („Properties“, Abb. 5.30), die die physikalischen Eigenschaften, wie Durchmesser, Länge, Druckverlustbeiwert, beschreiben und auch für die Erzeugung der ATHLET Eingabedaten benötigt werden.



Caption	Name	Value
X length	XL	0
Y length	YL	0.0
Z length (Geodetic height)	ZL	3
Left diameter	DL	0.5
Right diameter	DR	0.5
Left DEP0	DEPL	0.0
Right DEP0	DEPR	0.0
Length	L	3
Left cross sectional area	AL	0.0
Right cross sectional area	AR	0.0
Volume	V	0.58904862

Abb. 5.30 Properties für die Geometrie-Objekte in ATM

Aus den beschriebenen Basisobjekten lässt sich ein Leitungsstrang als ATHLET TFO, entweder als „Pipe“ oder „Branch“ – Objekt modellieren, entsprechend der Beschreibung im ATHLET User’s Manual /LER 12/. Analog gibt es in ATM entsprechende Objekte (siehe Abb. 5.31) dafür, die die wichtigsten Daten zur Beschreibung von ATHLET TFOs enthalten.

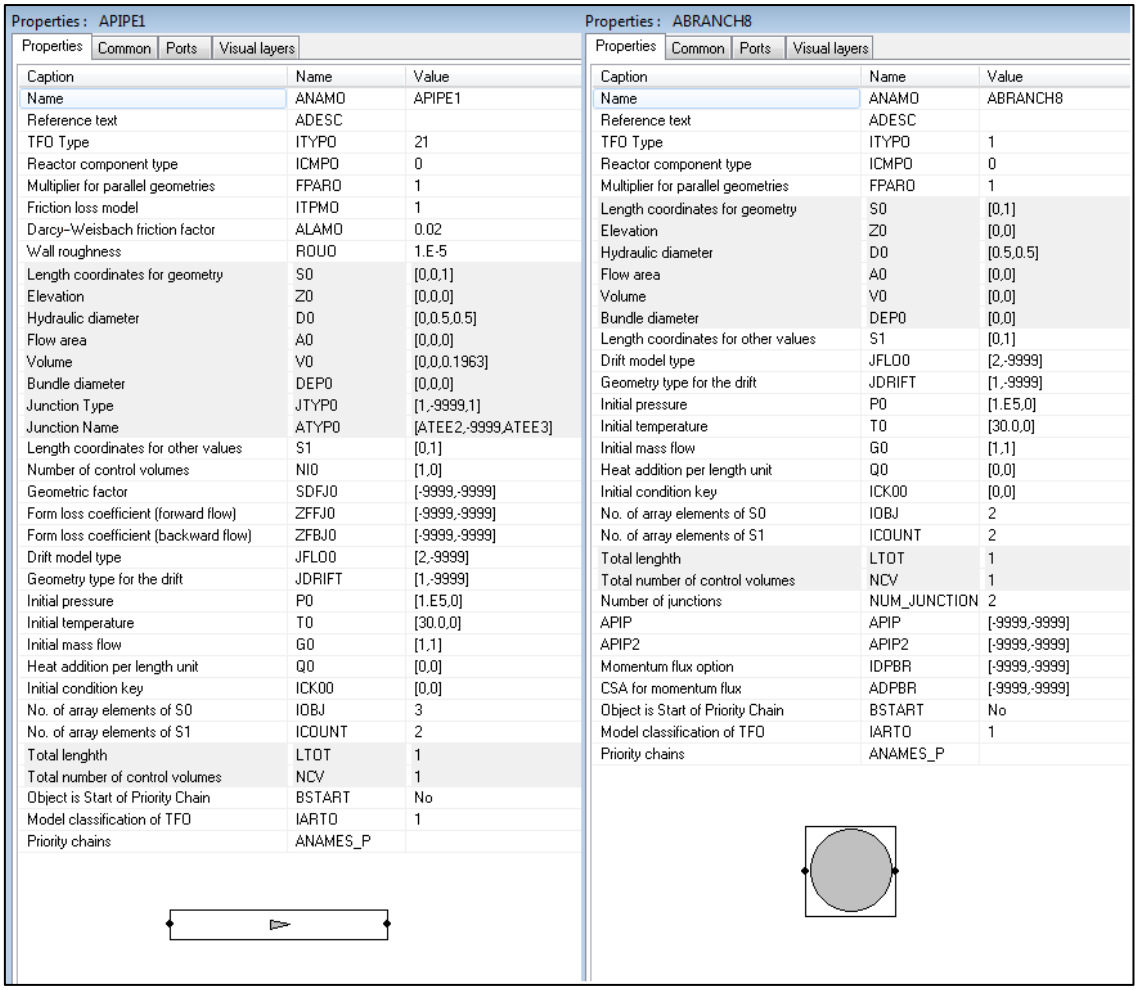


Abb. 5.31 ATHLET-Fluid-Objekte in ATM

Diese Objekte besitzen zur Geometriebeschreibung ein besonderes Unterdiagramm („Submodel“), das eine Kette von Unterobjekten enthält, aus denen die grau hinterlegten Daten generiert werden. In Abb. 5.32 ist eine fiktive Rohrleitung mit einem ATHLET Pipe modelliert. Die Rohrleitung ist von links nach rechts definiert aus Geometrieobjekten, beginnend mit einem Anschluss, direkt anschließend ist ein Einspeisung vorgesehen, danach ein vertikales Rohrstück, ein Drosselventil, eine Verzweigung, ein schräger Diffusor usw. Die Anschlüsse, Ventile und Pumpen werden auch in der schematischen Darstellung, rechts oben im Bild, automatisiert eingeblendet. In der Geometriedarstellung werden die Strömungsquerschnitte, Rohrlängen und geodätischen Höhen der Rohrobjekte maßstabgerecht angezeigt, für Rohrbögen ist vereinfacht nur der Bogenwinkel als Zahlenwert eingeblendet. Bei Veränderungen der Geometrieobjekte werden die entsprechenden Daten der TFOs (Abb. 5.31) automatisch angepasst.

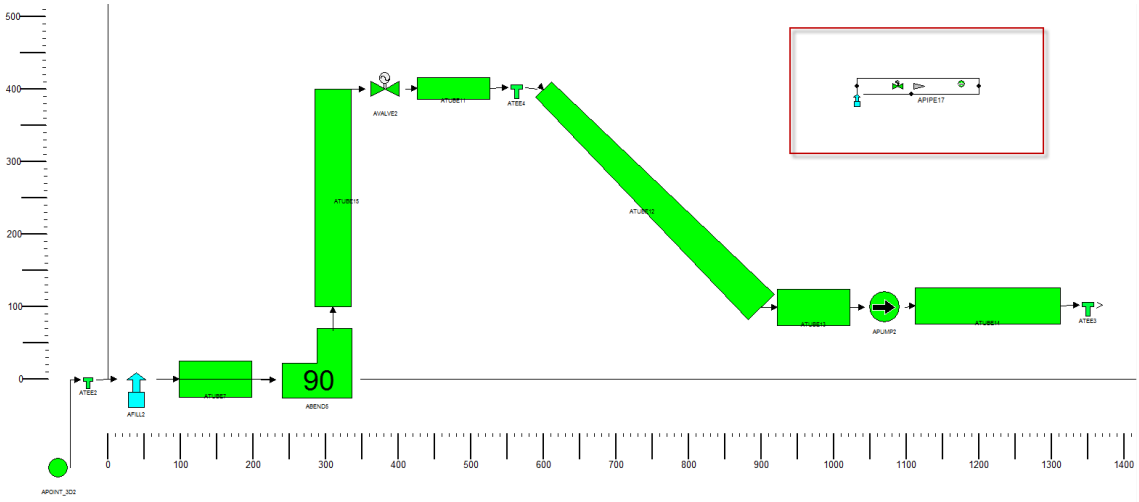


Abb. 5.32 Beispiel für ein ATHLET-Pipe mit Basisobjekten in ATM

Für die Eingabe der Metadaten der TFOs ist eine spezielle Dialogmaske (Abb. 5.33) für gekoppelte Tabellenwerte verfügbar, die über das Kontextmenü des Objekts angesprochen werden kann.

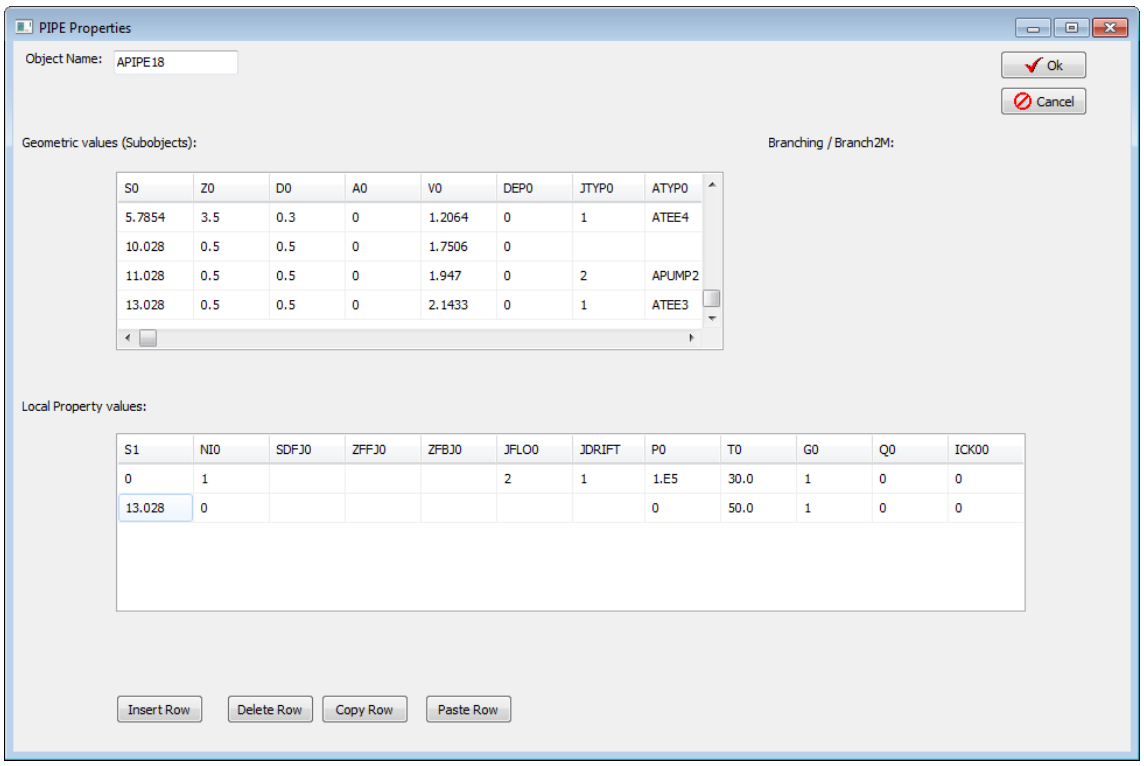


Abb. 5.33 Tabelle für lokale Daten der ATHLET-Fluid-Objekte in ATM

In der oberen Tabelle werden die aus den Geometrieobjekten abgeleiteten Daten angezeigt, können dort aber nicht verändert werden. Dies ist nur im Diagramm für das

Submodel des TFOs möglich. Weitere ortsabhängige Daten für das TFO, wie z. B. lokale Anfangswerte für Druck und Temperatur, die Einteilung in Zellen, Reibungsbeiwerte u.a. können in der unteren Tabelle für beliebige Längskoordinaten vorgegeben werden. Beim Beenden des Dialogs mit „OK“ werden die Eingaben in die Properties des TFO übertragen.

Das „Branch“-Objekt, das in ATHLET zur Modellierung von Verzweigungen, wie z. B. im oberen Plenum des RDB, verwendet wird, benötigt ähnliche Daten wie das „Pipe“-Objekt, kann aber keine internen Verbindungen haben. An einem „Branch“-Objekt können beliebig viele Leitungen (Pipe Objekte) angeschlossen sein. Die Anzahl und Lage der Anschlüsse kann interaktiv eingegeben („Number of junctions“, Abb. 5.31) werden, die Hauptströmungsrichtungen für die angeschlossenen Leitungen können im unter „Branching“ (Abb. 5.34) vorgegeben werden.

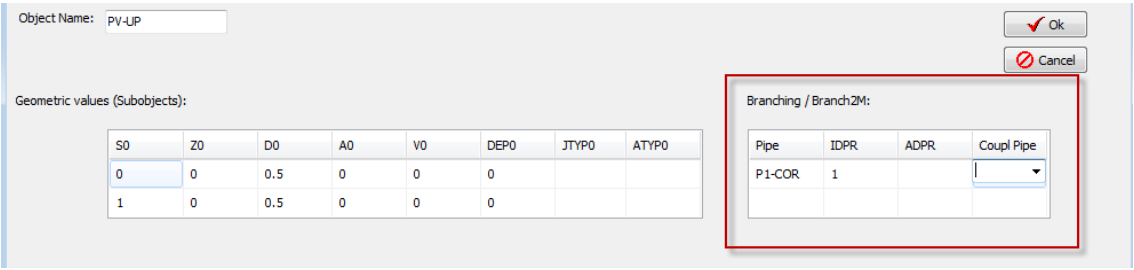


Abb. 5.34 Tabelle für „Branching“-Daten in ATM

Die Verwendung mehrerer Leitungs- und Knotenobjekte in einem Diagramm und ihre Verbindung mit Strömungslinien („Hydraulical Wires“) erlaubt die Modellierung eines Leitungsstrangs aus diesen Objekten. In Abb. 5.35 ist ein einfaches thermohydraulisches Modell für einen Kühlkreislauf in einem Druckwasserreaktor dargestellt, das 2 Branches und 7 Pipes verwendet.

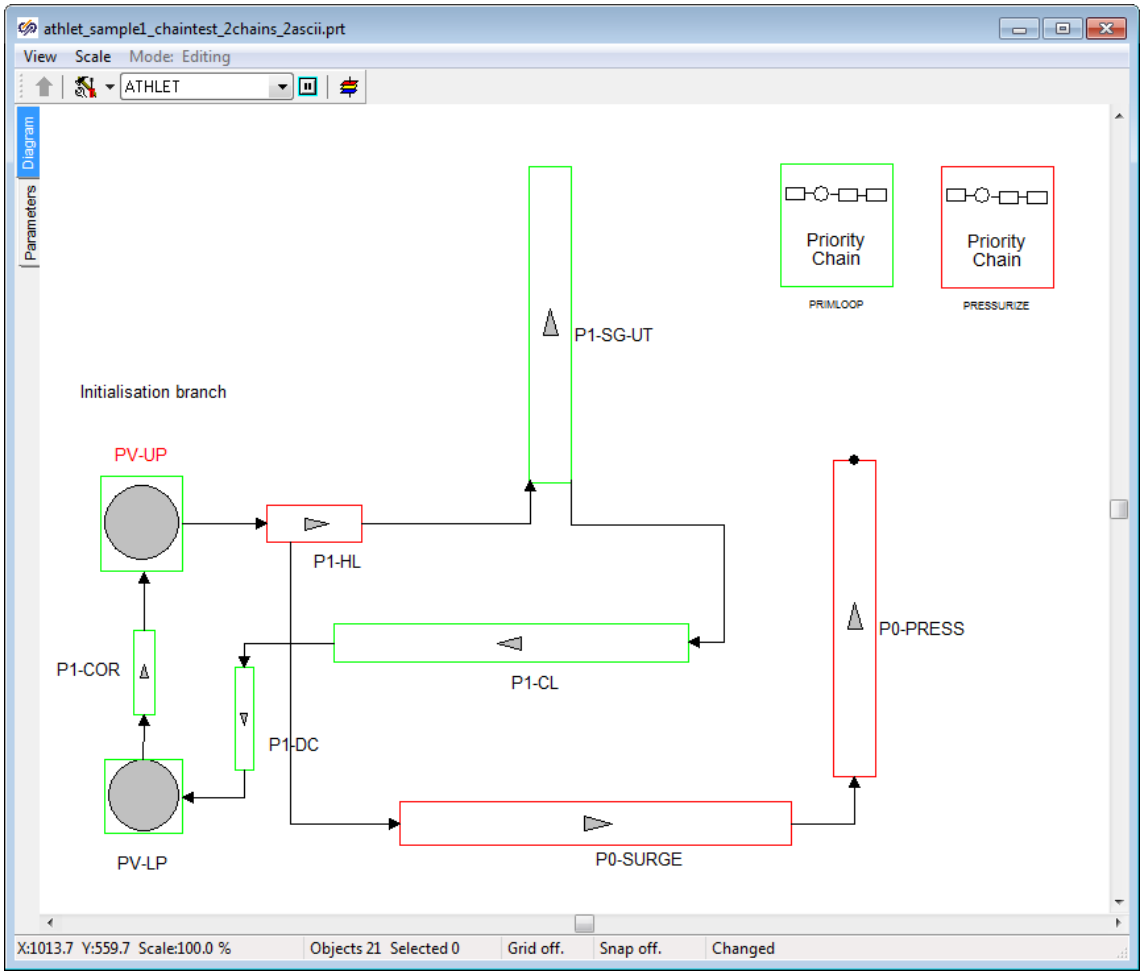


Abb. 5.35 Modellierung eines Kühlkreislaufs in ATM

In ATHLET werden die Verbindungen der Fluidobjekte in sogenannten Prioritätsketten definiert. Diese geben an, welche Objekte miteinander verbunden sind und definieren zusätzlich die Reihenfolge der Berechnung und die Position der Verbindung. In ATM sind zur Bereitstellung dieser Information „Priority Chain“-Objekte (PCO) vorhanden. Für jede in einem Modell benötigte Kette ist ein eigenes Objekt dieser Art nötig. Die Zuordnung von Fluidobjekten zu einer PCO ist durch die Selektion der gewünschten Objekte und das Kontextmenü der PC möglich (Abb. 5.36). Die Zugehörigkeit eines TFOs zu einer Prioritätskette ist durch die verwendete Konturfarbe erkennbar.

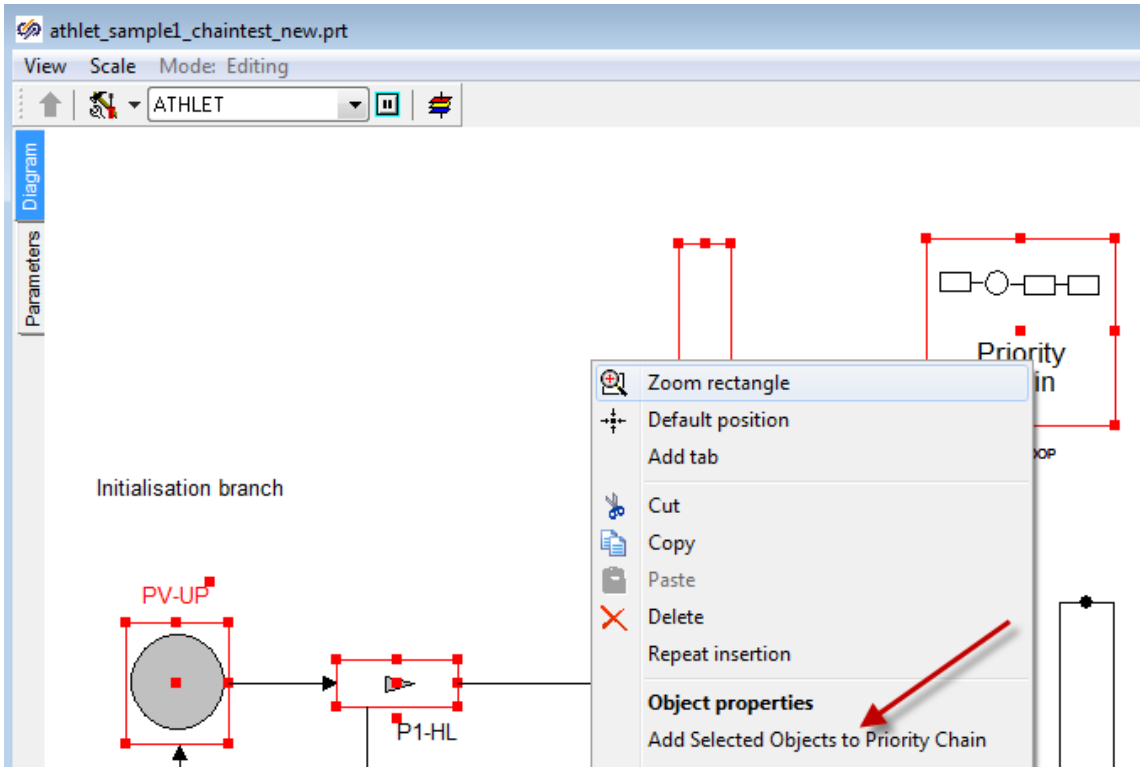


Abb. 5.36 Definition von Prioritätsketten in ATM

Bei der Bestimmung der Objekt-Reihenfolge wird in ATM die Richtung der Strömungsverbindung verwendet. Daher können keine Objekte zu einer bestehenden Kette hinzugefügt werden, wenn das vorangehende Objekt nicht bereits in der Kette enthalten ist. Ein Beginn einer neuen Kette ist hingegen, jederzeit möglich, wie es auch ATHLET erlaubt. Die Ortskoordinaten der Verbindungen werden direkt aus der Geometriedefinition der Pipe-Objekte („Tees“) übernommen.

Nachdem alle Objektdaten, ihre Verbindungen und die Prioritätsketten vorgegeben sind, kann ein äquivalenter ASCII-Datensatz für die Verwendung in der ATHLET Eingabe erzeugt werden. Der Name der Datei kann in den „Simulation Properties“ des Diagramms (Abb. 5.37) angegeben werden, die mit dem Icon  angezeigt werden können.

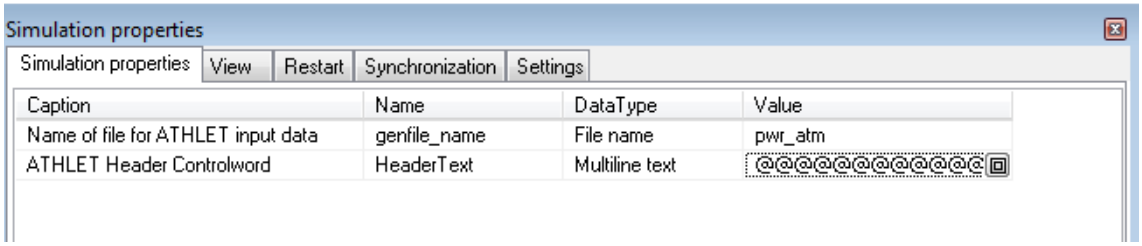


Abb. 5.37 Namensvorgabe für ATHLET-ASCII-Datei in ATM

Die Prozedur zur Ausgabe der Daten arbeitet alle vorhandenen Prioritätsketten- und Fluidobjekte nacheinander ab. Dabei wird geprüft, ob alle Verbindungen („Ports“) belegt sind und die notwendigen Daten eingetragen sind. Eventuelle Fehler werden in der Statuszeile des Diagramms aufgeführt.

Die Erstellung eines Thermohydrauliknetzwerks wurde an verschiedenen Beispielen, getestet. Für das Primärkreislaufmodell eines DWR (Abb. 5.35), der in einem Beispieldatensatz von ATHLET, verwendet wird, konnte gezeigt werden, dass bei Vorgabe identischer Daten, der von ATM erzeugte Datensatz dieselben Ergebnisse wie der Beispieldatensatz liefert.

Das vorliegende ATM-Prototypsystem erlaubt die interaktive Eingabedatenerstellung für eine Reihe von Daten für thermohydraulische Netzwerke. Es ist möglich, mit dem vorliegenden Entwicklungsstand einen Teil der im ATHLET-Datensatz benötigen Daten für die TFOs zu erzeugen. Eine Präsentation des Prototyps und Diskussionen mit Modellentwicklern und Anwendern haben jedoch ergeben, dass der Prototyp noch erheblicher Anpassungen im Hinblick auf eine effiziente Nutzbarkeit bedarf. Außerdem fehlt die Unterstützung einer Vielzahl von ATHLET-Modellen, insbesondere für die Wärmeleitobjekte.

6 Methoden zur Auswertung und Visualisierung von Simulationen

Die durchgeführten Arbeiten stellen Methoden bereit zur Visualisierung und Auswertung von Felddaten mehrdimensionaler Simulationen der thermohydraulischen und neutronenphysikalischen Vorgänge im Reaktor und berücksichtigen die besonderen Anforderungen der Reaktorsicherheitscodes ATHLET, QUABOX/CUBBOX und TORT.

Aus der Vielzahl verfügbarer Anwendungs-Software, sowohl kommerziell als auch OpenSource, wurde ein geeignetes Werkzeug ausgewählt, das die vielschichtigen Anforderungen wie flexible Darstellungsoptionen, Importmöglichkeiten von CAD- und Gitterdaten, Darstellung zeitabhängiger Daten, Programmierschnittstellen und zeitgemäßen Bedienoberflächen erfüllt.

Ein einheitliches, standardisiertes Datenformat für alle in der Visualisierung benötigten Daten wurde zur Speicherung der dynamischen Ergebnisdaten und der Geometrie- und Nodalisierungsinformationen festgelegt. Die Erzeugung dieser Daten aus den Daten im CAD-System und den Simulationsdaten der Codes wird durch Konvertierungsprogramme und eine Bedienoberfläche unterstützt.

Verschiedene Darstellungsmöglichkeiten der Daten wurden untersucht und dokumentiert, die die modellspezifischen Bedürfnisse der betrachteten Simulationsmodelle berücksichtigen. Zur Vereinfachung der Bedienung des Visualisierungssystems wurden typische Anwendungsfälle zur Visualisierung von ATHLET-Simulationen erfasst und die dafür auszuführenden Bedienungsschritte festgehalten.

Als Anwendung und Test des Gesamtsystems wurde ein Modell eines Reaktordruckbehälters mit den anschließenden Leitungen im Visualisierungssystem dargestellt und die Daten aus einer mehrdimensionalen Störfallsimulation eingespielt und angezeigt.

6.1 Erfassung des Stands und Anforderungen bei Auswertung und Visualisierung mehrdimensionaler Simulationen

6.1.1 Spezifische Anforderungen der GRS-Codes

Die Anforderungen der GRS-Codes zur 3D-Ergebnisdarstellung und -Analyse sind heterogen. Es lassen sich grob zwei Klassen bilden: Software, die bereits dreidimensional rechnet und Simulationsprogramme, die dreidimensionale Systeme mit eindimensionalen Gleichungen modellieren. Zur ersten Gruppe gehören Neutronenkinetikprogramme, wie QUABOX/CUBBOX und TORT-TD ebenso wie die Fluiddynamik-Codes CFX und OpenFOAM, deren Ergebnisse auf einem 3D-Gitter berechnet und dabei entweder dessen Punkten oder Zellen zugeordnet werden. In die zweite Gruppe fallen z. B. ATHLET und COCOSYS: diese Codes bauen auf benannten, gekoppelten Objekten auf, in die das Simulationsgebiet zerlegt wird, und welche zusätzlich noch eindimensional in Kontrollvolumen (control volume nodes) unterteilt sein können. Neben einer geometrischen Auswahl von Regionen ist es in diesen Fällen also wichtig, auch eine Adressierung über Namen zu ermöglichen.

Die meisten Codes berechnen neben skalaren Größen, die sich mit Hilfe von Farbkodierungen abbilden lassen, auch vektorielle Ergebnisvariablen. Im Fall von ATHLET ergibt sich die Besonderheit, dass durch die Entwicklung eines 3D-Moduls die Anforderung besteht, nicht nur die berechneten 1D-Flüsse zwischen den Objekten und in ihrem Inneren abzubilden, sondern auch die resultierenden dreidimensionalen Vektorgrößen. Weil in ATHLET selbst dafür nur unzureichende geometrische Informationen vorliegen, können diese Daten dort nicht berechnet werden. Eine weitere Besonderheit von ATHLET liegt darin, dass nicht alle Variablen in allen Objekten definiert sind.

Für Simulationen im Bereich der Reaktorsicherheitsanalyse wird eine gute Unterstützung der Visualisierung zeitlicher Phänomene und der dabei unter Umständen anfallenden großen Datenmengen benötigt. Dazu können verschiedene Maßnahmen beitragen, wie selektiver Import aus Multi-Block-Datensätzen, bedarfsgesteuerter Datenzugriff (streaming), sowie die Ausnutzung von mehreren Prozessoren zur Darstellung (parallel rendering), oder allgemein sogar mehreren (entfernten) Computern (rendering farm, remote visualisation).

6.1.2 Visualisierung und Analyse

Außer einer interaktiven 3D-Ansicht werden von den gängigen Darstellungs- und Manipulationsmöglichkeiten in Visualisierungssystemen vor allem Ausschnitte (clipping), Schnittebenen (slicing) und Isokonturen (contours) für polygonale Repräsentationen von skalaren Feldern, sowie Glyphen- (glyphs), und Stromlinien-Abbildungen (streamlines) von Vektorfeldern für die GRS-Codes gebraucht. Benannte Objekte sollten selektiv einblendbar sein. Für die Ergebnispräsentation sind das Abspeichern von Bildern (idealerweise sowohl im Bitmap- als auch in Vektorformaten) und die Erstellung von Animationen wichtig.

Programmübergreifend erfordern Szenarienanalysen oft den direkten Vergleich von zwei oder mehr Rechenläufen. Dieser Anwendungsfall sollte am besten ebenso unterstützt werden wie das Speichern von häufig benutzen Auswertungskonfigurationen (sessions).

Zusätzlich zu einer qualitativen Bewertung von Simulationen über 3D-Ansichten müssen auch quantitative Analysen unterstützt werden. Unter diese Anforderung fallen numerische, tabellarische (spreadsheet) und selektive Datenansichten (probing), 2D-Graph-Darstellungen von zeitlichen oder räumlichen Variablenverläufen (charts), und die Erstellung von abgeleiteten Größen.

6.1.3 Generische Auswahlkriterien

Außer den rein technischen Fähigkeiten der in Frage kommenden Visualisierungsprogramme gehen auch generische Kriterien zur Bedienbarkeit und zur Nachhaltigkeit der Software ein. Die Bedienbarkeit einer Software umfasst einerseits ergonomische Aspekte wie eine intuitive Benutzerführung, gute Dokumentation und informative Anwenderforen, aber auch Flexibilität und Interoperationsfähigkeit, was durch Makros, die Zugänglichkeit der Programmfunktionen aus einer Skriptsprache und über Datenschnittstellen oder sogar direkte Anbindungen von anderen Spezial- oder Simulationsprogrammen hergestellt werden kann. Die herangezogenen Nachhaltigkeitskriterien betreffen Aspekte die dazu beitragen, dass die Software auch in Zukunft mit vertretbarem Aufwand benutzbar gehalten werden kann. Dieser Aspekt wird von verschiedenen Faktoren wie der weltweiten Größe und Aktivität der Benutzer- und Entwicklerbasis, der Plattformunabhängigkeit (mindestens Linux und Windows), der Verwendung von

etablierten Standards, der Aussicht auf Weiterentwicklung und der Erweiterbarkeit durch Anwendercode, z. B. über eine Plugin-Schnittstelle, bestimmt.

Nachhaltigkeitskriterien spielen auch eine Rolle bei der Frage, ob ein kommerzielles oder eine quelloffenes (OpenSource) Programm in Frage kommt. Eine OpenSource-Lösung bietet neben den entfallenden Anschaffungskosten und dem vollständigen Zugriff auf den Quellcode den Vorteil, dass auch zukünftige Weiterentwicklungen von Dritten genutzt werden können. Nachteile gegenüber kommerziellen Anwendungen ergeben sich unter Umständen aus mangelnder Dokumentation und durch die fehlende, kommerzielle Unterstützung bei der Anwendung des Programms durch die Entwickler, was aber durch eine große und aktive Anwendergemeinde ausgeglichen werden kann. Die Gefahr, dass die Entwicklung eines Programms eingestellt wird besteht dagegen sowohl bei kommerziellen als auch quelloffenen Programmen. Weil die GRS einige ihrer Programme auch Dritten zur Verfügung stellt, sollten die Anschaffungskosten für die Auswertesoftware auf jeden Fall in einem vertretbaren Rahmen bleiben.

6.1.4 Datenformate

Die Auswahl der Visualisierungssoftware ist eng gekoppelt an die Frage nach dem zu benutzenden Datenformat, die deswegen parallel dazu untersucht werden muss. Neben den angeführten Anforderungen bezüglich der Unterstützung von, skalaren und vektoriellen Punkt- und Zelldaten, benannten Objekten mit inhomogenen Variablen und großen, zeitabhängigen Datenmengen sollte die Speicherung in einem Binärformat erfolgen, das plattformunabhängig, gut dokumentiert und nicht proprietär ist und außerdem von vielen Programmen unterstützt wird. Wegen der unter Umständen sehr großen Anzahl der Zeitschritte in ATHLET ist es wünschenswert, dass alle Informationen in *einer* Datei abgelegt werden können. Eine wichtige Eigenschaft ist auch die Erweiterbarkeit des Datenformats, die automatisch gegeben ist, wenn es auf Metaformaten wie HDF5 /HDF 97/ oder NetCDF /NET 90/ aufsetzt.

6.2 Vergleich und Auswahl geeigneter Visualisierungswerkzeuge

6.2.1 Marktübersicht

Die Einarbeitung in eine komplexe Visualisierungssoftware nimmt einige Zeit in Anspruch. Deshalb wurde zunächst versucht, eine Marktübersicht und eine erste Einschätzung aus den über Internet zugänglichen Beschreibungen von verschiedenen kommerziellen (EnSight, FieldView, AVS/Express, Avizo) und quelloffenen Programme (ParaView, VisIt, OpenDX) zu erlangen. Nach den vorliegenden Informationen bieten alle Programme die oben geforderten grundlegenden Visualisierungsmöglichkeiten und die skriptgesteuerte Automatisierung. Von der Software OpenDX wurde allerdings schon seit 1997 keine neue Bedienungsanleitung mehr veröffentlicht und der letzte Eintrag auf der Web-Seite stammt von 2007, so dass man davon ausgehen muss, dass keine Weiterentwicklung mehr stattfinden. Im Gegensatz zu den anderen Kandidaten ist OpenDX auch nur eingeschränkt plattformunabhängig und scheidet damit aus. Ebenso fällt Avizo durch das Anforderungsraster, weil es die Aufteilung von Gittern in Objekte anscheinend nicht unterstützt. Da die Lizenzen für die kommerziellen Produkte recht teuer sind, z. B. EnSight-Standard mit 4000$ pro Jahr und Lizenz, und regelmäßig erneuert werden müssen, wurde die Auswahl im Sinne der angelegten generischen Kriterien auf die verbliebenen Open-Source-Programme eingeschränkt. Kostenlose Versionen kommerzieller Programme, etwa von Ensight, existieren zwar, stellen aber durch ihre Limitationen, z. B. der Anzahl der darstellbaren Objekte, keine nachhaltige Lösung dar.

6.2.2 Praxis-Test und Auswahl

ParaView und VisIt entstammen beide dem Umfeld amerikanischer Großforschungseinrichtungen. Sie beruhen auf der 3D-Grafik- und Bildverarbeitungsbibliothek VTK (Visualization Toolkit), deren Entwicklung 1993 bei GE Corporate Research initiiert, und die heute von der Firma Kitware Inc. betreut wird. Kitware zeichnet auch zusammen mit dem Los Alamos National Laboratory (LANL) und den Sandia National Labs für die Programmierung von ParaView verantwortlich. VisIt wird unter anderem vom Lawrence Livermore National Laboratory (LLNL) und vom Oak Ridge National Laboratory (ORNL) entwickelt. Beide Programme sind auf die Verarbeitung großer zeitaufgelöster Datenmengen ausgelegt, unterstützen den Betrieb von entfernten Visualisierungsclustern und können Sessions mit vordefinierten Datenverarbeitungs-

prozessen anlegen. Als Skriptsprache dient jeweils Python; die Programmiersprache ist C++, die auch für die Erstellung von benutzerdefinierten Erweiterungen (plugins) benutzt wird. ParaView und VisIt werden häufig aktualisiert und erfüllen formal auch die verbleibenden spezifischen und generischen Anforderungen.

Im Benutzerbetrieb stellte sich heraus, dass die Programme in ihrer Hauptfunktionalität sehr ähnlich sind. VisIt verwendet als Standard-Datenformat Silo und hatte zum Zeitpunkt des Tests leichte Vorteile bei der quantitativen Analyse (VisIt: query) und bei der Auswahl von Datenbereichen (selection). Allerdings wurde die Benutzeroberfläche von ParaView wegen des Einhaltens von etablierten GUI-Standards subjektiv als erheblich intuitiver empfunden, weil sie den Anwender leitet und weitestgehend versucht, Bedienungsfehler schon im Vorfeld zu vermeiden. Des Weiteren unterstützt ParaView für sein Standard-Datenformat EXODUS II /EXO 95/ nicht nur die Aufteilung des Gitters in Objekte (blocks), die außerdem selektiv eingelesen werden können, sondern zusätzlich deren hierarchische Anordnung in sogenannte "assemblies". Alle in VisIt verfügbaren Eingabeformate, unter anderem Silo, können auch im Konkurrenzprogramm über eine standardisierte Wrapper-Schnittstelle eingelesen werden. Der Silo-Standard ist zwar flexibel genug, auch mehrere Zeitschritte in einer Datei zu speichern, allerdings unterstützen VisIt und damit auch der ParaView-Importfilter nur einen Zeitschritt pro Datei.

Sowohl ECODUS II als auch Silo können auf dem HDF5-Format aufsetzen. Das Silo-Format ist allerdings so komplex, dass eine Software-Bibliothek benötigt wird, um Dateien zu erzeugen. Obwohl auch für EXODUS II eine Bibliothek existiert, ist der Standard konzeptionell so einfach, dass die direkte Erzeugung über eine HDF5/NetCDF-Schnittstelle möglich ist. Durch Beseitigung von Abhängigkeiten reduziert sich die Komplexität von Software, wodurch sie wartungsfreundlicher wird.

Neben den bereits genannten Gründen spricht für ParaView außerdem, dass sowohl die in der GRS verwendete Software OpenFOAM als auch die im Kerntechnik-Bereich bekannte Simulationsplattform Salome dieses Programm als Standard-Postprozessor einsetzen. Darüber hinaus wird kommerzieller Support von der Firma Kitware Inc. angeboten. Aufgrund der Erfüllung der dargelegten Entscheidungskriterien und des erfolgreichen Anwendungstests fiel die Wahl deshalb vorzeitig auf ParaView, ohne dass die verbliebenen kommerziellen Konkurrenten näher untersucht wurden.

Für das Datenformat kamen neben EXODUS II und dem bereits aussortierten Silo außerdem noch CGNS, ein Standard aus der Fluiddynamik, und XDMF in Frage. XDMF

speichert Daten in einer Kombination aus HDF5 und XML. Aufgrund von mangelnder Dokumentation, fehlender institutioneller Unterstützung und seiner damit ungewissen Zukunft wurde es nicht weiter betrachtet. Letztendlich wurde EXODUS II gegenüber CGNS wegen seiner besseren Unterstützung in ParaView der Vorzug gegeben.

6.3 Entwicklung von Schnittstellen und Funktionen zum Datenimport

6.3.1 Konzept

Das Konzept für die 3D-Visualisierung von ATHLET-Ergebnissen fügt sich nahtlos und modular in die bestehende ATHLET-Verarbeitungskette ein, die also unabhängig davon auch weiterhin angewendet werden kann (Abb. 6.1), die bisher genutzten Elemente sind hellblau unterlegt): ein geometrisches 3D-Modell wird mit FreeCAD erzeugt, mit Objekt- und Nodalisierungsinformationen versehen und in einer Datei im Wavefront OBJ-Format /OBJ 89/ zwischengespeichert. Optional kann eine Datenübertragung der für ATHLET relevanten geometrischen und topologischen Informationen zu dem Präprozessor ATM (siehe Kap. 5.5) erfolgen. Ohne den Datentransfer an ATM ist der Ersteller des Eingabedatensatzes selbst für die Konsistenz von 1D- und 3D-Daten verantwortlich. Das Programm obj2exii konvertiert die OBJ-Datei in das EXODUS II-Format. Nach erfolgter Simulation werden die ATHLET-Ergebnisdaten mit einem weiteren Werkzeug, pd2exii, ausgelesen und in die erstellte EXODUS II-Datei geschrieben, die sich mit ParaView zur Auswertung öffnen lässt. Die nur schwache Kopplung der beteiligten Programme über Dateien gewährleistet die angestrebte hohe Modularität und damit im Bedarfsfall die einfache Austauschbarkeit der einzelnen Komponenten.

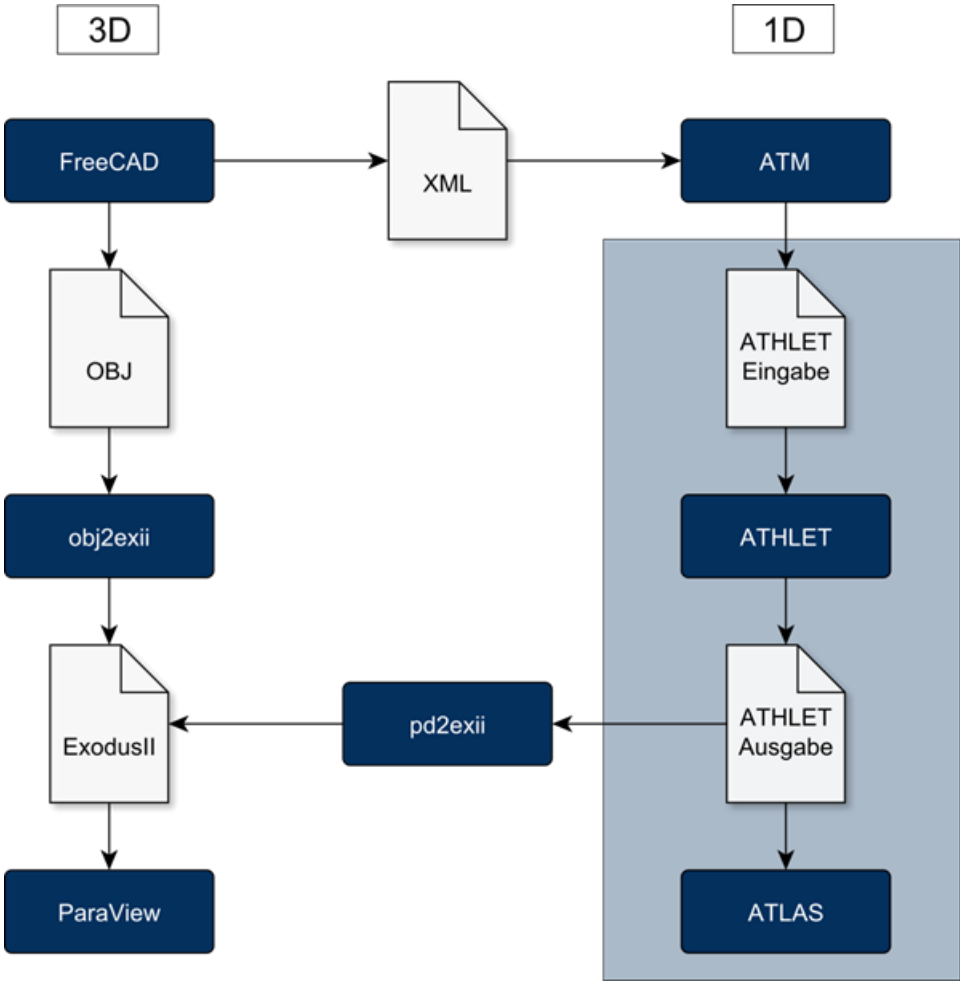


Abb. 6.1 Ablaufdiagramm der ATHLET-Verarbeitungskette

6.3.2 Umsetzung

Für das Einlesen der benötigten ATHLET-Ergebnisdaten einerseits und zur Behandlung von 3D-Daten andererseits wurden die Python-Pakete "records" und "eph3dr1nes" /PYT 15/ programmiert. Sie enthalten sowohl wiederverwendbare Bibliotheksfunktionen als auch darauf beruhende Werkzeuge, insbesondere die Konvertierer obj2exii und pd2exii, die Geometrie-Präprozessoren athlet3dcylinder, athlet3dgrid und ihre gemeinsame grafische Benutzeroberfläche.

6.3.2.1 obj2exii

Die meisten Geometrie generierenden Programme, darunter auch FreeCAD, bieten OBJ als ASCII-Exportformat an. Es erlaubt die Unterteilung von Geometrie in disjunkte

Objekte und ist eingeschränkt über die Speicherung von Zusatzinformationen in Kommentarzeilen erweiterbar. Geometrische Formen werden über ihre in Dreiecke zerlegten Oberflächen beschrieben. Eine Zeile der OBJ-Datei enthält entweder eine mit dem Buchstaben "o" eingeleitete Objektdefinition, der der Objektname folgt, eine Punkt- (Buchstabe "v"=vertex) oder eine Zellangabe (Buchstabe "f"=face). Ein Punkt wird durch drei kartesische Koordinaten, eine Dreieckszelle durch die Angabe von drei von eins aufwärts gezählten, globalen Punktindizes angegeben. Kommentarzeilen beginnen mit einer Raute. Jede Zelle gilt als Bestandteil des zuletzt definierten Objektes. Punkte können zu mehreren Objekten gehören. Die hier verwendeten Bestandteile stellen nur eine Untermenge des eigentlich weitaus umfangreicheren OBJ-Standards dar.

Weil OBJ keine Objekthierarchien kennt, setzt obj2exii, voraus, dass jedes ATHLET-Kontrollvolumen (node, Knoten) von dem CAD-Programm in ein eigenes OBJ-Objekt geschrieben, und dabei der ATHLET-Objektname durch eine konfigurierbare Trenn-Zeichenkette und den Volumenindex ergänzt wird. Obj2exii übersetzt diese Strukturen in das EXODUS II-Format, das auf dem generischen NetCDF aufsetzt. Benannte Objekte können in EXODUS II in Form von Element-Blöcken (element blocks) gespeichert werden. In den Zeilen der entsprechenden NetCDF-Felder (connect*) stehen die zugehörigen Zelldefinitionen. Die zweite Felddimension richtet sich nach dem über ein NetCDF-Attribut definierten Zell-Typ des Blocks (hier: TRIANGLE, TETRA oder SPHERE). Die kartesischen Punktkoordinaten aller in den Blöcken enthaltenen Zellen befinden sich separiert in drei globalen eindimensionalen Feldern (coordx, coordy, coordz). Obj2exii setzt die ATHLET-Objekte wieder aus ihren Kontrollvolumen zusammen und legt die Nodalisierungsinformationen in Form von Zellengrenzen in NetCDF-Attributen der entsprechenden Blöcke ab. Diese speziellen Attribute sind zwar nicht im EXODUS II-Standard vorgesehen, zerstören aber aufgrund des Namen basierten Zugriffs auf NetCDF-Dateien das Format nicht und werden im nächsten Arbeitsschritt zur Zuordnung der Daten benötigt.

Obj2exii erzeugt bis zu drei verschiedene EXODUS II-Dateien. Neben den gerade diskutierten Kontrollvolumen werden optional auch ATHLET-Flussgrößen (junction data) separat gespeichert. Zu diesem Zweck berechnet obj2exii die Mittelpunkte der Kontrollvolumen; gegebenenfalls liest es sie stattdessen als Meta-Informationen aus OBJ-Kommentaren. Diese Punkte, die zur weiteren Bearbeitung durch pd2exii auch in Nicht-Standard-EXODUS-Felder (centers_eb*) geschrieben werden, bilden, objektweise zusammengefasst, die geometrischen Speicherorte für die aus den ATHLET3D-

Daten zu berechnenden, integrierten 3D-Vektoren. Die Koordinaten der Punkte für die ursprünglich von ATHLET ausgegebenen 1D-Flussgrößen bestimmt obj2exii aus den Schnittpunkten der Verbindungsvektoren topologisch benachbarter Kontrollvolumen mit ihren Oberflächendarstellungen. Die Zellen der Element-Blöcke in den Vektor-Dateien haben den Typ "SPHERE".

Da die Geometrien der ATHLET-Objekte zunächst nur in einer Oberflächendarstellung vorliegen, sind auf Schnittbildern in ParaView Hohlräume zu sehen. Möchte man eine ausgefüllte Darstellung erreichen, müssen die Kontrollvolumen nach dem Einlesen mittels 3D-Delaunay-Triangulierung von der Dreiecksbeschreibung in eine Zerlegung mit Tetraeder-Zellen überführt werden. Diesen optionalen Schritt delegiert obj2exii mit Hilfe des Python-Moduls „ctypes“ an die als Bibliothek verwendete Software TetGen /TET 15/, die Hang Si am Weierstrass-Institut in Berlin entwickelt.

6.3.2.2 pd2exii

Das Programm pd2exii liest für jeden Zeitschritt die binären ATHLET-Ausgabedaten ein und weist sie anhand der Namen der in der EXODUS II-Datei erwähnten Objekte den entsprechenden Block eigenen Datenstrukturen zu. Im Gegensatz zu den Kontrollvolumendaten (vals_elem_var*), die an EXODUS-Zellen gebunden sind, legt pd2exii Flussgrößen als Punktdaten ab (vals_nod_var*). Über ein dafür vorgesehenes EXODUS-Feld (elem_var_tab bzw. nod_var_tab) können Variablen, die in einem Objekt nicht vorkommen, gekennzeichnet werden. Benannte geometrische Objekte, die keine Entsprechung in dem ATHLET-Modell haben, verwenden den gleichen Mechanismus um zu verhindern, dass ein lesendes Programm überhaupt nach Variableneinträgen sucht.

Die Richtungen der 1D-Flussgrößen sind zeitlich konstant. Sie entsprechen den normierten Verbindungsvektoren der beiden jeweils betroffenen Kontrollvolumenmittelpunkte, wobei die Orientierung sich aus der von ATHLET vorgegebenen Hauptflussrichtung ergibt. Daneben müssen nur die von ATHLET berechneten, dynamischen Skalarfaktoren als Variablen in der EXODUS-Datei vorgehalten werden. Im Gegensatz dazu sind die 3D-Vektoren auch in ihrer Richtung zeitlich variabel, weil sie sich jeweils aus den 1D-Flüssen eines Kontrollvolumens zusammensetzen. Es gibt verschiedene Möglichkeiten, die entsprechenden 1D-Vektoren zu kombinieren. Pd2exii berücksichtigt hierfür in jedem Zeitpunkt nur die gerade ausströmenden Beiträge. Ana-

log zu den 1D-Vektoren werden sowohl Richtungsvektor als auch Skalarwert in unterschiedlichen EXODUS II-Variablen gespeichert, allerdings jeweils beide für jede Flussgröße. Außerdem wird die Orientierung jetzt bereits im Vektor kodiert und bei dem Skalar handelt es sich um den integralen Absolutbetrag.

Bei der Konvertierung der ATHLET-Kontrollvolumendaten bildet pd2exii jeweils einen Wert auf mehreren Zellen (Dreiecke oder Tetraeder) ab. Je nach Detaillierungsgrad der Geometrie werden Informationen also mehr oder weniger redundant abgelegt. Zur Vermeidung dieses Problems, das zu sehr großen Dateien führen kann, bietet pd2exii die Option einer effizienteren Speicherung, die aber eine Nachverarbeitung im Visualisierungsprogramm erfordert. Zu diesem Zweck werden die Kontrollvolumendaten in den Knoten-Mittelpunkten objektweise als EXODUS II-Punktmengen (node sets) abgelegt. Die Element-Blöcke enthalten in diesem Fall nur die Geometrie.

6.3.2.3 athlet3d{grid,cylinder}

Die hier dargestellte Visualisierungslösung wurde in ihrer Entwicklung stark durch die Anforderungen des parallel in RS1507 entwickelten 3D-Moduls für ATHLET getrieben. Um die Zeitspanne zu überbrücken, in der noch keine generische Lösung für die Erstellung von 3D-Geometrien existiert, wurden speziell auf ATHLET3D zugeschnittene Präprozessoren für einfache kartesische und zylindrische Gitter geschrieben. Die erstellten Objekte folgen einer vorgegebenen, ihre Position kodierenden Namenskonvention und beziehen optional verfügbare 1D-Geometrieangaben wie Höhe und Länge von ATHLET. Die Programmausgabe besteht jeweils aus einer OBJ-Datei als Eingabe für obj2exii.

6.4 Entwicklung und Anpassung von Visualisierungsoptionen und der Bedienoberflächen

6.4.1 GUI

Über die programmierte grafische Benutzeroberfläche lassen sich die Werkzeuge zur Datenkonversion und zur Erstellung von einfachen Modellgeometrien für ATHLET3D einheitlich bedienen (siehe Abb. 6.2). Sie ist ebenfalls in Python geschrieben und basiert auf der mit dem Interpreter mitgelieferten Fenster-Bibliothek Tkinter, so dass die

Software-Abhängigkeiten gering gehalten werden. Die GUI verfügt über eine einfache Sitzungsverwaltung, die die Parameter erfolgreich durchgeführter Programmläufe lokal speichert. Einzelne Anwendungsprogramme werden als Python-Module eingeladen und ihre Hauptprozedur (main) über ein assoziatives Array mit den GUI-Eingabedaten bestückt. Die Kommunikation der Werkzeuge mit dem Anwender erfolgt über ein Informationsfenster, in das die Logging-Ausgabe jeweils umgeleitet wird. Als Hilfe stehen dem Benutzer automatisch generierte HTML-Seiten mit den Optionen der einzelnen Programme über Menü-Einträge zur Verfügung. Außerdem existieren ein Benutzer- und ein Entwicklerhandbuch für das Python-Paket.

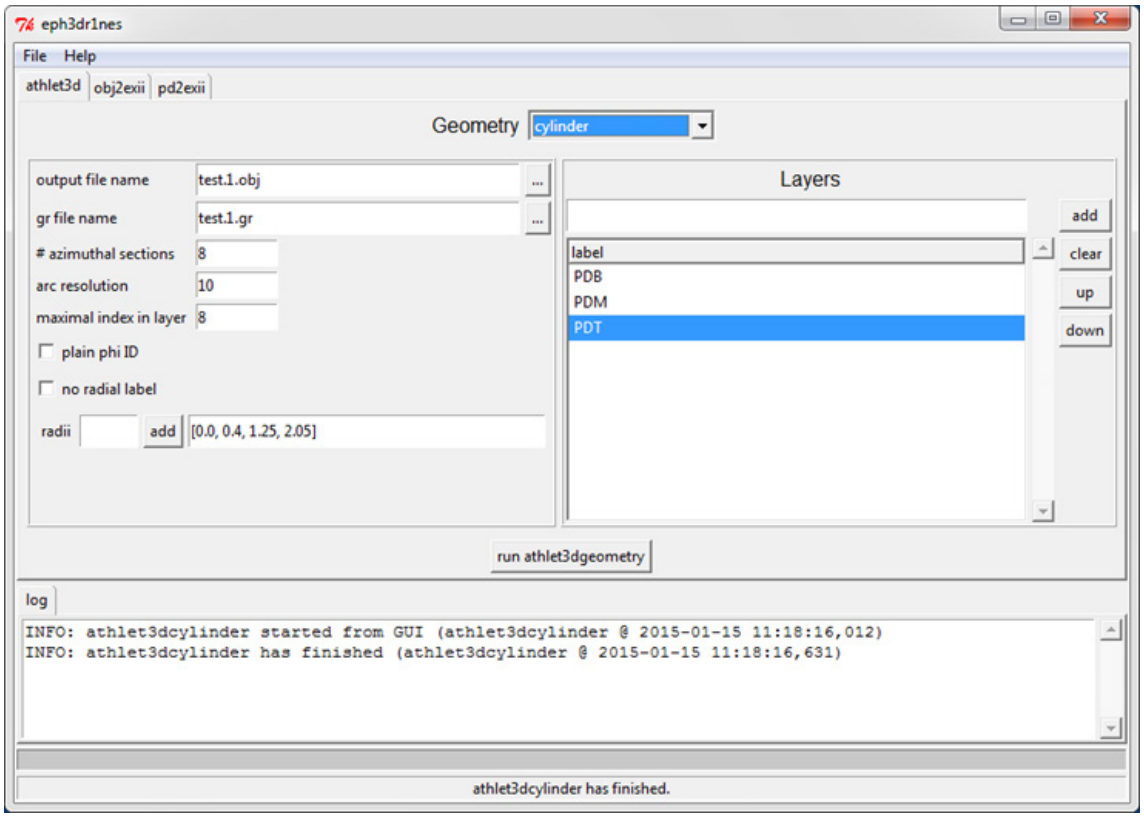


Abb. 6.2 GUI für die Konvertierungs- und Geometrie-Werkzeuge aus dem EPH3DR1NES-Paket

6.4.2 ParaView

Wie in Abschnitt 6.3.2.2 erwähnt, muss für die Darstellung von mit pd2exii erzeugten nicht-redundanten EXODUS II-Dateien eine Nachverarbeitung erfolgen, um jede Zelle eines Element-Blocks mit dem numerischen Wert des zugehörigen ATHLET-Kontrollvolumens, der in einer entsprechenden Punktmenge gespeichert ist, zu versehen. Zu diesem Zweck dient ein in Python programmierbarer Filter in ParaView. Er

verwendet Informationen über die Zellintervalle der geometrischen Modelle der Kontrollvolumen, die als Zelldaten in jeder Punktmenge vorliegen. Weil in der Benutzeroberfläche von ParaView vor der Anwendung des programmierbaren Filters mehrere Optionen manuell zu aktivieren sind, wurde ein ParaView-Makro erstellt, das die Ausführung dieser Arbeitsschritte und die Anwendung des Filters auf einen Mausklick reduziert, wodurch sich Benutzerfehler reduzieren lassen.

Eine Modifikation der ParaView-GUI zur Vereinfachung von typischen Aufgaben bei der Auswertung von ATHLET-Daten wurde nicht vorgenommen. Eine bessere Lösung ist die Erstellung von standardisierten Auswertungen in Form von ParaView-Zustandsdateien. Beim Einladen einer Zustandsdatei hat der Anwender die Möglichkeit die Speicherorte aller enthaltenen Datenquellen anzugeben, wodurch die Auswertungen wiederverwendbar werden. Dadurch sind flexible, benutzerdefinierte, für jedes Programm individuelle Anpassungen ohne Programmierung möglich.

6.5 Anwendung des Werkzeugs für die Visualisierung typischen 3D-Störfallsimulationen

Zum Test der 3D-Visualisierung wurde für ATHLET ein Datensatz für einen Druckwasserreaktor mit detaillierter Darstellung des RDB mit 17 Kernkanälen (Abb. 6.3) und 4 Kühlkreisläufen mit vereinfachter Modellierung der Sekundärseiten gewählt.

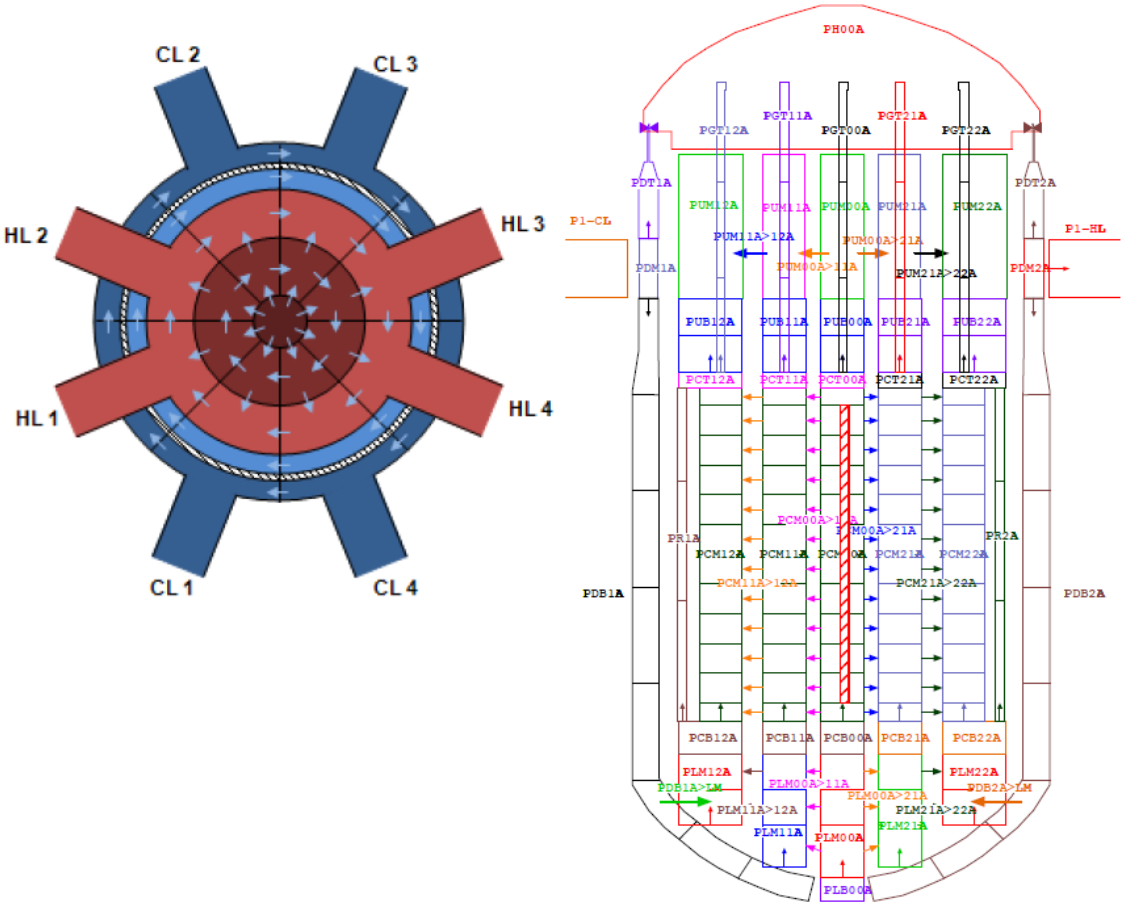


Abb. 6.3 Schematische Darstellung der Nodalisierung im RDB (radial und axial)

Als Transiente wurde die Simulation des Ausfalls einer einzelnen Hauptkühlmittelpumpe durchgeführt. Dieser Störfall wurde gewählt, da sich in seinem Verlauf unsymmetrische Fluidzustände in den Kreisläufen und im Kern ausbilden.

Eine realistische Simulation des Falls benötigt die Nachbildung der Eingriffe der Reaktorbegrenzungssysteme und des Frischdampfsammlers, die im vorliegenden Datensatz nicht vorhanden war. Eine vereinfachte Ergänzung dieser Systeme erfolgte auf Basis einer Referenzsimulation mit dem Analysesimulator einer Konvoi-Anlage. Die Simulation wurde dann mit den folgenden Randbedingungen durchgeführt:

- Einleitung der Transiente durch Abschalten der Hauptkühlmittelpumpe (HKP) in Kreislauf 1
- Leistungsreduktion durch Steuerstäbe (externe Reaktivität aus Referenzlauf)
- Vorgabe der Speisewassermassenströme (Tabellen aus Referenzlauf)
- Vorgabe der Dampfwassermassenströme

- Getrennte Berechnung der Dampfwassermassenströme durch Druckregelung der Dampferzeuger (da kein Sammler) mit einem einfachen Regler auf einen Sollwert von 75 bar.

Ohne die Berücksichtigung einer Druckregelung, die vereinfacht wie in Abb. 6.4 mit einer PI-Regelung modelliert wurde, ergeben sich falsche Sekundärtemperaturen, was dazu führt, dass auch die Kreislauftemperaturen auf der Primärseite unrealistisch simuliert werden. Für die Druckregelung wurden in verschiedenen Testläufen geeignete Reglerparameter ermittelt. Mit den verwendeten Randbedingungen und der Druckregelung konnte schließlich eine realistische Simulation im Vergleich mit der Referenzrechnung erzielt werden.

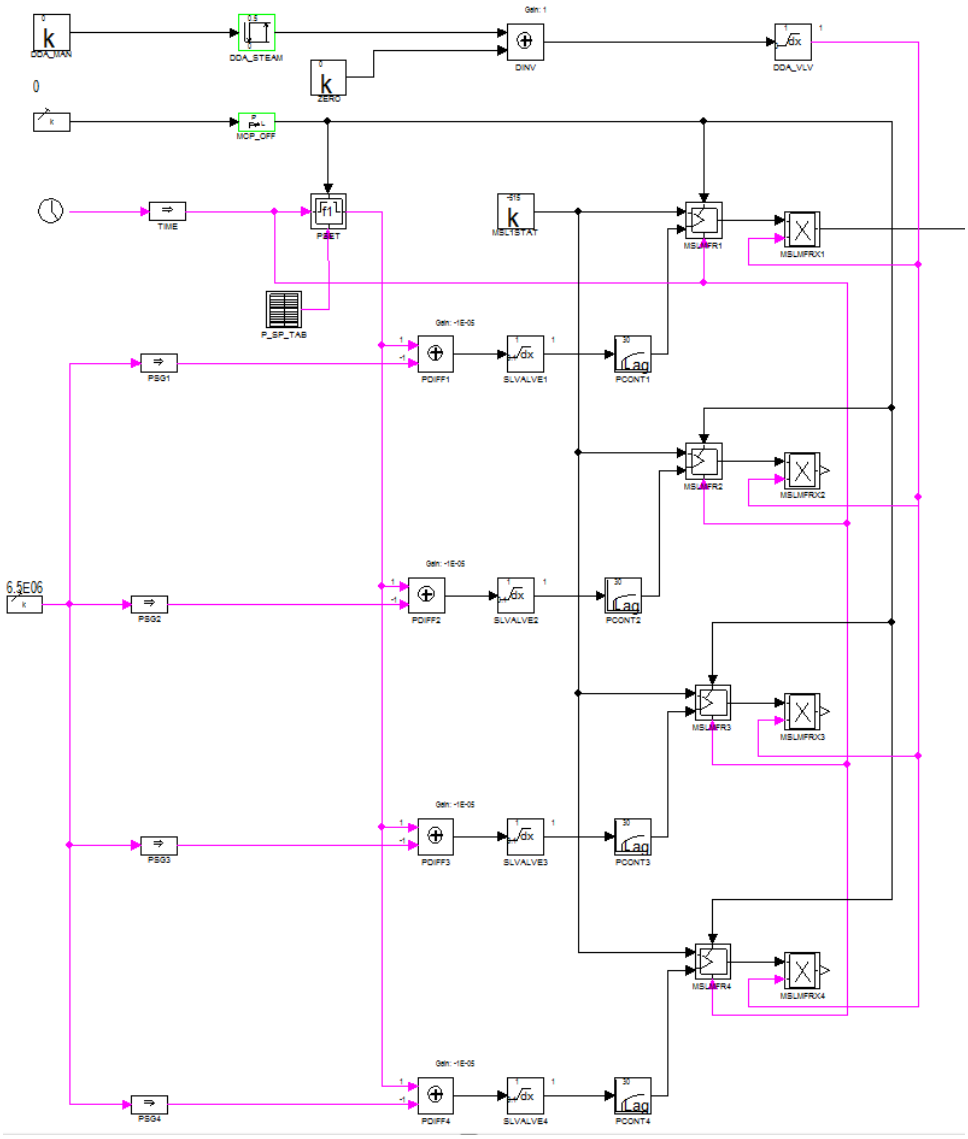


Abb. 6.4 Modellierung der Dampferzeuger-Druckregelung

Im CAD-System wurde ein 3D-Modell der Kühlkreisläufe und des RDB mit einer dem Datensatz entsprechend Diskretisierung erstellt. Die Daten wurden exportiert und zusammen mit den dynamischen Ergebnisdaten mittels der Python-Konvertierungsprogramme in das EXODUS II- Format überführt. Die erzeugten Daten wurden direkt in ParaView verwendet, um verschiedene Visualisierungsmöglichkeiten zu testen.

Die dynamischen Temperaturverteilungen im RDB und den Kreisläufen wurden unter Verwendung verschiedener Farbskalen dargestellt. Eine Skala von Blau nach Rot (Abb. 6.5) zeigt intuitiv kalte und heiße Bereiche, eine Regenbogenskala (Abb. 6.6) hat leichte Vorteile bei der Unterscheidbarkeit der Werte. Eine Ausblendung von einzelnen Objekten (Teile des RDB in der linken Ansicht) oder die Anpassung der Schnittebenen (Längsschnitt durch den RDB in der rechten Ansicht) ermöglichen einen guten Einblick in die axialen und radialen Temperaturverteilungen im RDB.

Die Visualisierung der Massenströme durch Vektoren zeigt die Entwicklung der Strömungsumkehr im Kreislauf mit der defekten Pumpe, die im Bild durch ein „X“ gekennzeichnet ist. Die Länge der Vektoren ist ein Maß für die Stärke der Strömung entsprechend der lokalen Fluidgeschwindigkeiten. Eine gleichzeitige Darstellung mit den Temperaturen ist möglich, wenn mit durchsichtigen Volumina („Transparenz“) gearbeitet wird.

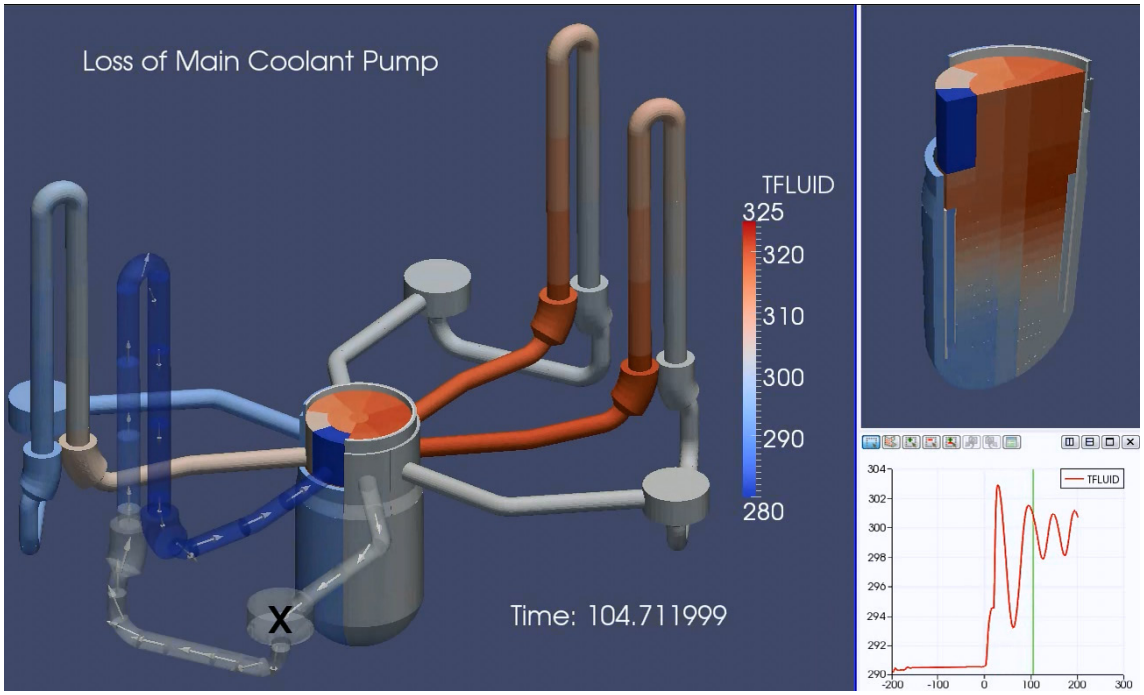


Abb. 6.5 Ausfall einer HKP in einem DWR (Rot-Blau-Farbskala)

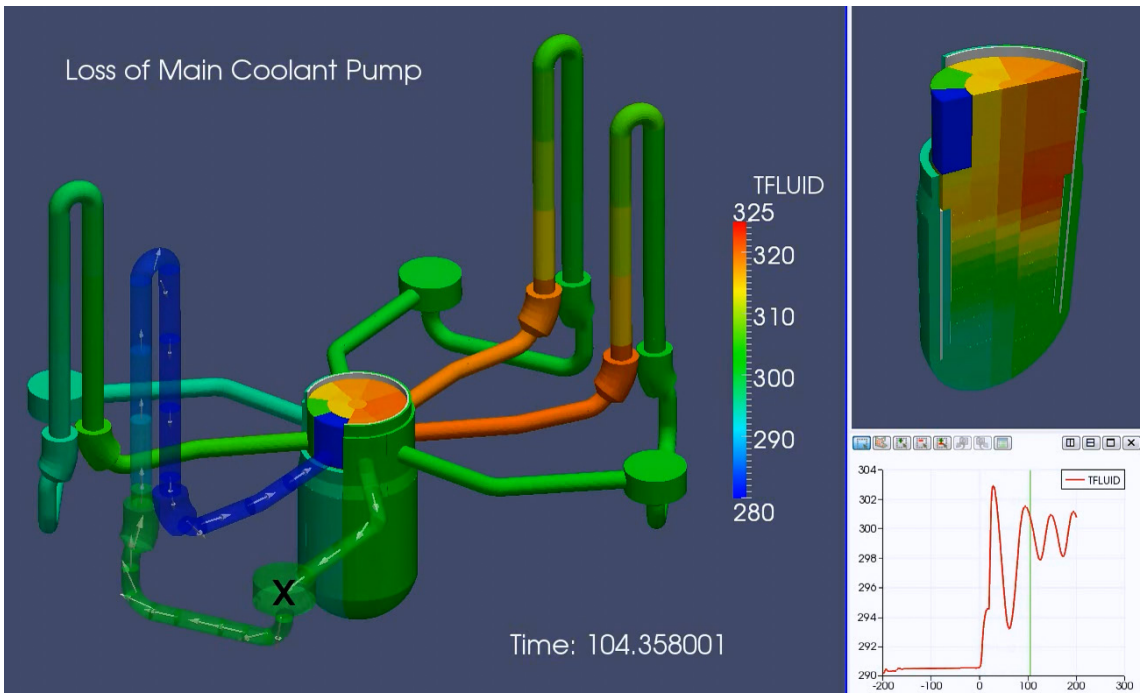


Abb. 6.6 Ausfall einer HKP in einem DWR (Regenbogen-Farbskala)

In ParaView können Datenbereiche entsprechend des ATHLET-Gitters mit den Namen der Thermofluidobjekte ausgewählt und gekennzeichnet werden. Die Identifizierung der ATHLET-Objekte in der 3D-Visualisierung ist möglich, indem die Geometrie gemäß

ihrer Nodalisierung eingefärbt oder mit Text-Annotationen versehen wird. Damit wird es erleichtert, gewünschte Datenbereiche zu lokalisieren und bei Bedarf Zusatzinformationen wie den lokalen Zahlenwert oder den zeitlichen Verlauf (Temperaturverlauf im defekten Kreislauf in der Ansicht rechts unten) anzuzeigen.

In einer weiteren Anwendung wurde die Simulation eines Steuerstabauswurfs in ParaView visualisiert. Die lokalen Stableistungen wurden mit dem Neutronendynamikcode QUABOX/CUBBOX berechnet. Die Geometrie des Simulationsgitters ist im Gegensatz zum vorigen Fall ein einfaches kartesisches Gitter, das programmatisch für ParaView erzeugt werden kann. In der gezeigten Darstellung (Abb. 6.7) ist die lokale Leistung entsprechend einer blau-roten Farbskala dargestellt. Es wird ein spezielles Visualisierungsverfahren, das „Volume Rendering“ /WIK 15/ benutzt. Dies ist eine Technik, die den verschiedenen Bildbereichen („Voxels“) eine gewisse Transparenz zuordnet und damit Bilder ähnlich wie mit einem Röntgengerät erzeugen kann. Die Visualisierung zeigt mit diesem Verfahren sehr gut, dass sich für den vorderen Bereich des Kerns (Position des ausgeworfenen Steuerstabs) eine, im Vergleich zu weiter hinten liegenden Kernbereichen, erhöhte Leistung einstellt. Zusätzlich zum Reaktorkern ist noch ein statisches Modell des Druckbehälters eingeblendet, das sowohl Transparenz als auch Reflexion einer virtuellen Lichtquelle in der Darstellung verwendet.

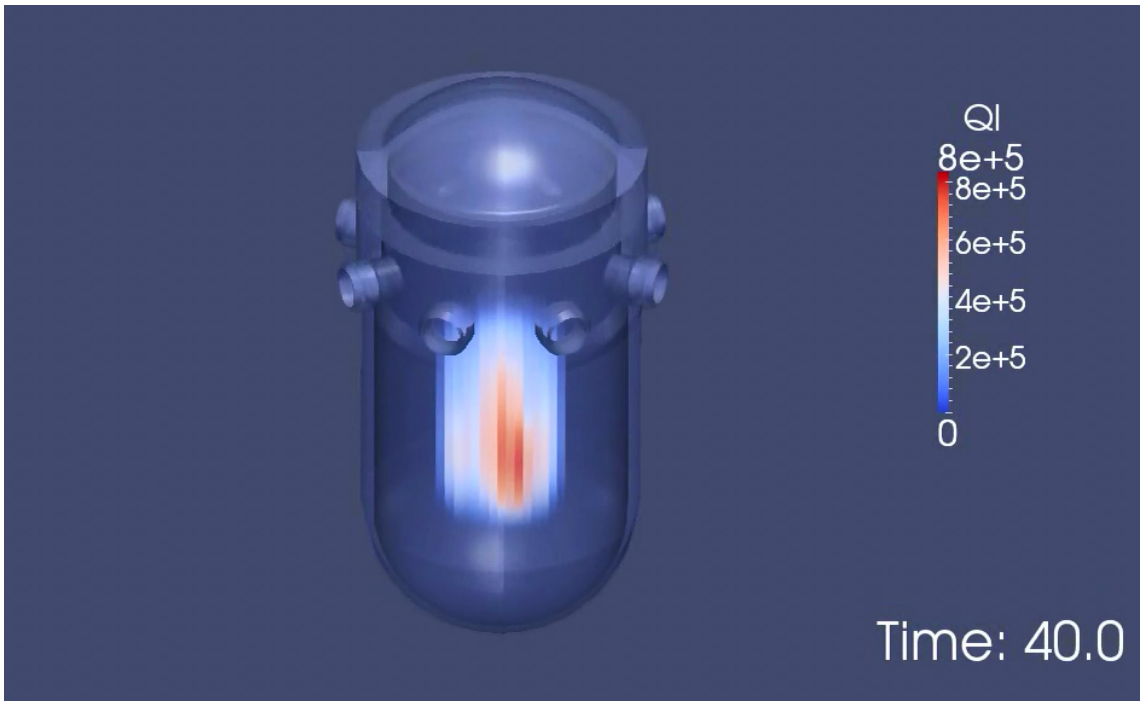


Abb. 6.7 3D-Visualisierung eines Steuerstabauswurfs mit Volume Rendering

6.6 Weitere Anwendungsbeispiele

Für die Entwickler des ATHLET3D-Moduls gab es bis zur Entwicklung der 3D-Visualisierungslösung für ATHLET-Daten nur eine technisch sehr eingeschränkte und aufwendige Methode auf der Basis von ATLAS, Ergebnisse zu visualisieren und zu verifizieren. Deswegen war insbesondere die neue integrierte 3D-Vektor-Darstellung in Verbindung mit der Möglichkeit, Gitter für einfache Modellgeometrien parametrisch zu erzeugen, für die Fehlererkennung und -behebung hilfreich. Umgekehrt flossen wertvolle Erfahrungen aus der Anwendung wieder in die Entwicklung der Visualisierungswerkzeuge zurück. In Abb. 6.8 und Abb. 6.9 sind jeweils Beispiele für die Verwendung der Geometriepräprozessoren durch ATHLET3D-Anwender und -Entwickler dargestellt.

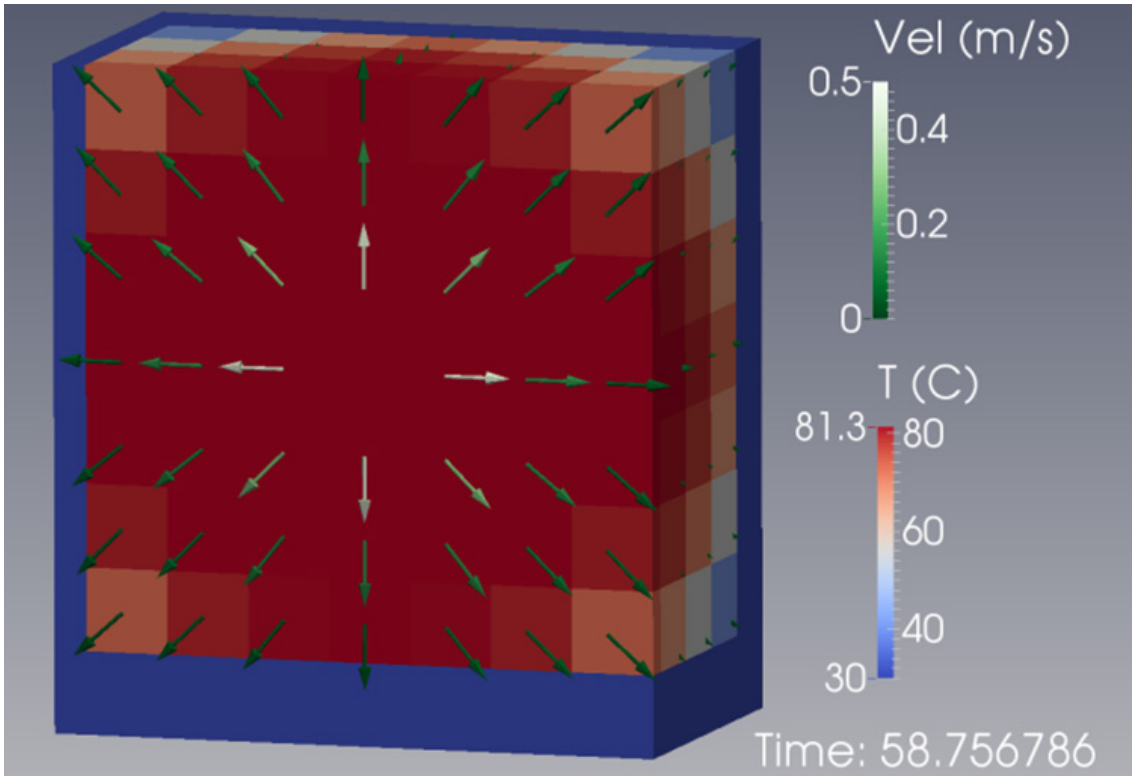


Abb. 6.8 ATHLET3D-Rechnung auf einem kartesischen Gitter (athlet3dgrid)

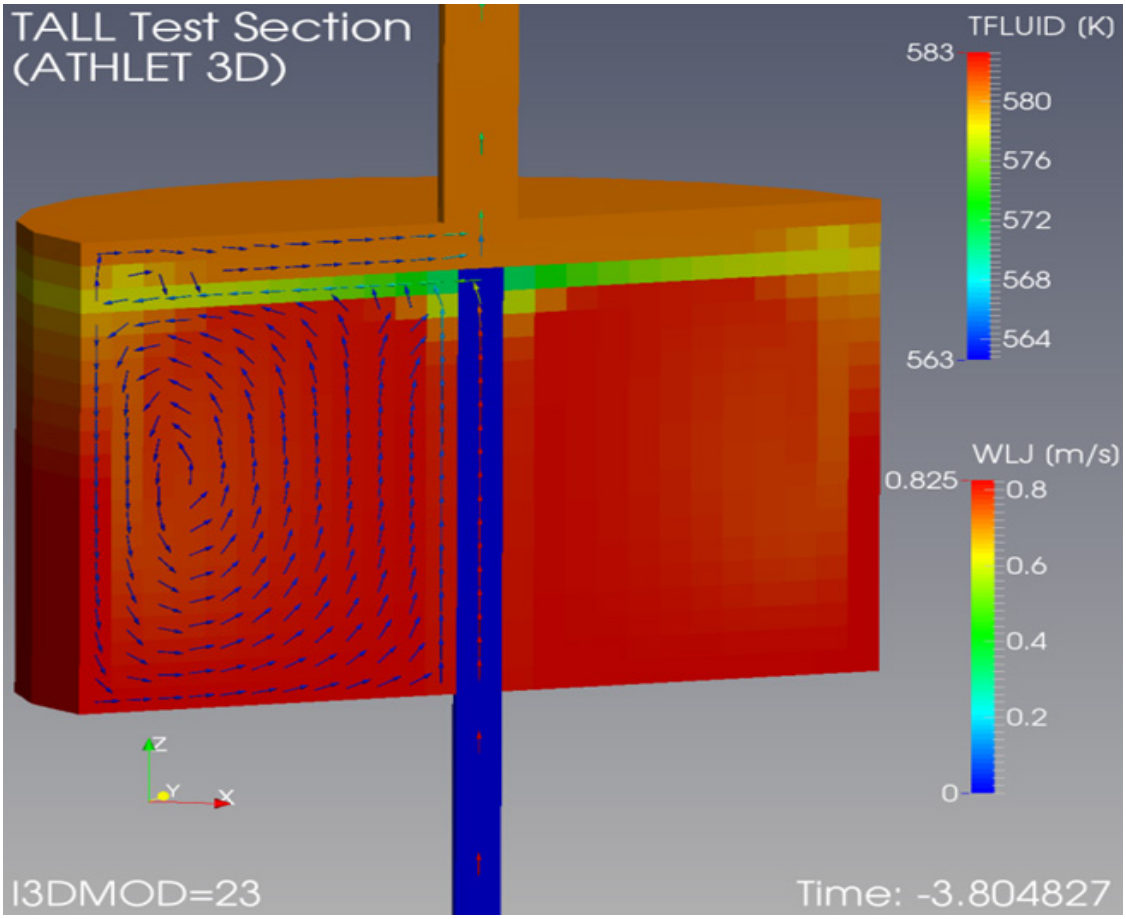


Abb. 6.9 ATHLET3D-Rechnung auf einem zylindrischen Gitter (athlet3dcylinder)

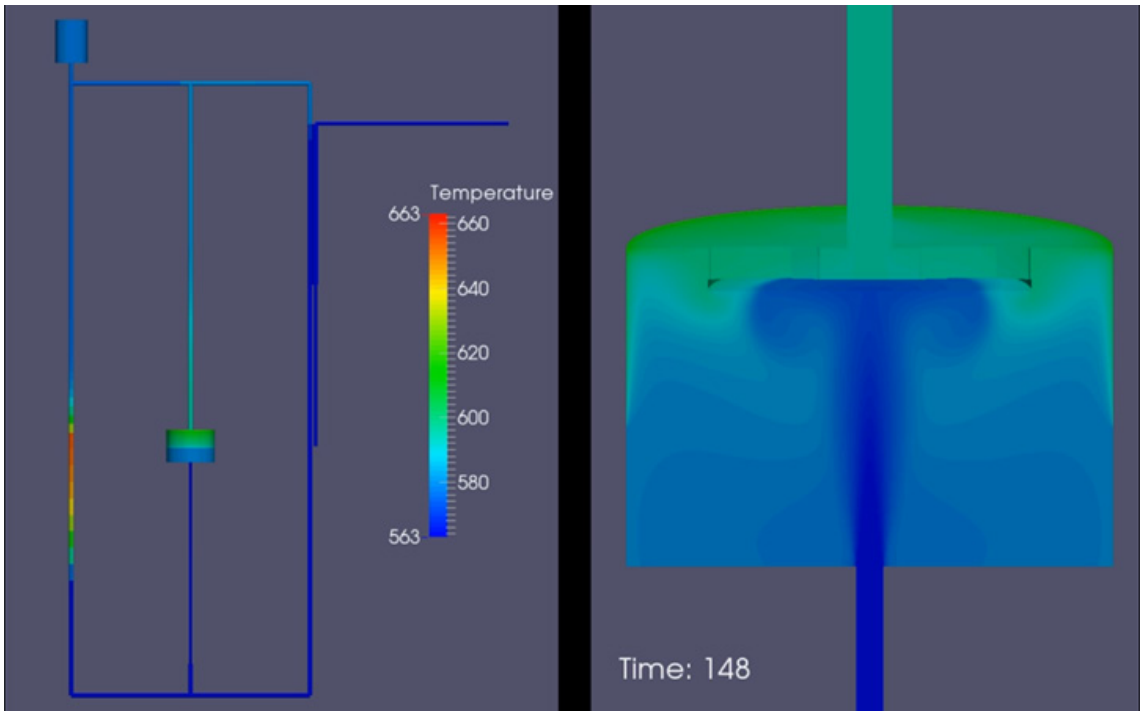


Abb. 6.10 Beispiel für die Visualisierung einer gekoppelten CFX-ATHLET-Rechnung

Auch für gekoppelte ATHLET-Rechnungen hat sich die 3D-Ergebnisvisualisierung schon bewährt. Ein Transiente in einem thermohydraulischen Versuchsstand wurde von einem Anwender zu einem Teil mit CFX und zum anderen Teil mit ATHLET modelliert. Weil sich die CFX-Daten durch Export in ein von ParaView lesbares Format bringen lassen, konnten erstmalig die Resultate beider Rechencodes gemeinsam dargestellt werden, nachdem für ATHLET ein geometrisches Modell mit den in diesem Projekt entwickelten Methoden in FreeCAD erstellt worden war (Abb. 6.10). Dieser Anwendungsfall war damit auch der erste Funktionsnachweis der Verarbeitungskette von der Geometrieerstellung bis zur Visualisierung. Ähnliche Synergieeffekte durch die Verwendung von standardisierten Datenformaten sind natürlich auch mit jedem anderen Programmsystem möglich, das an ATHLET gekoppelt ist und über entsprechende Ausgabemöglichkeiten verfügt, wie z. B. OpenFOAM, oder sogar direkt in den ATHLET-Ausgabevektor schreiben kann, z. B. QUABOX/CUBBOX.

7 Erweiterungen des Analysesimulators ATLAS

Der Analysesimulator ATLAS wird für ein großes Anwendungsspektrum, für viele Anlagentypen und von einem großen Benutzerkreis einsetzt. In diesen Anwendungen wird der Simulationsumfang ständig erweitert, insbesondere zur Berechnung lokaler Effekte wie z. B. „Clogging“ bei Leckstörfällen oder der Simulation neuer digitaler Leittechnik. Die hier beschriebenen Arbeiten tragen den zusätzlichen Anforderungen an die Visualisierungsmöglichkeiten in ATLAS Rechnung. Neu verfügbare Darstellungen in ATLAS sind einfache 3D-Objekte, GCSM-Diagramme aus AGM und Shapiro Diagramme zur Wasserstoffverbrennung. Dazu wurden die Bildbeschreibungssprache „APG“ von ATLAS erweitert und Funktionen zum Transfer der AGM Diagramme nach APG entwickelt. Die Shapiro Diagramme können automatisch für alle Codes mit Containmentsimulation (COCOSYS, ASTEC und MELCOR) erzeugt werden. Mit der neuen Option zum Vergleich mehrerer Simulationsrechnungen bzw. mit Messdaten wurde die Anwendbarkeit in der Analyse verbessert. Alle Änderungen wurden in einer neuen Release Version zusammengefasst und an die Anwender verteilt.

7.1 ATLAS-Bilder mit 3D-Objekten

Die langjährige Erfahrung mit ATLAS hat gezeigt, dass für die meisten Anwendungen die Verwendung zweidimensionaler Anlagendarstellungen eine zufriedenstellende Auswertung gewährleistet. Eine 3D-Visualisierung ist aber bei der Darstellung von Komponenten, die räumlich hintereinander angeordnet sind vorteilhaft, wie z. B. die Brennstäbe in einem Brennelement, die Brennelemente eines Reaktorkerns, oder die 4 Kühlkreisläufe eines Druckwasserreaktors. Solche Fälle kommen häufig bei Bildern vor, die durch die ATHLET Input Graphic (AIG) erzeugt werden. Aus diesen Gründen wurde die Einführung dreidimensionaler Objekte in ATLAS realisiert.

7.1.1 APG-Befehle für die 3D-Geometrie

Zur Abbildung dreidimensionaler Objekte war es erforderlich, die Bildbeschreibungssprache „APG“ von ATLAS durch eine Reihe neuer Befehle zu erweitern, bei denen im räumlichen Bezugssystem außer X und Y auch die Z-Koordinate angesprochen wird. Diese Befehle werden durch den Zusatz „3D“ gekennzeichnet. Zuerst bestand die Notwendigkeit, den Bereich des geometrischen Raums zu definieren, in dem die drei-

dimensionalen Objekte gezeichnet werden sollen. Zu diesem Zweck wurde der Befehl „AXIS3D“ eingeführt. Diesem Befehlswort folgen 6 Gleitkommazahlen (xmin, xmax, ymin, ymax, zmin, zmax), welche die Grenzen des geometrischen Raumabschnittes festlegen, z. B.

AXIS3D -0.1 5.0 0.0 7.0 -3.0 4.0

Ist der Raumabschnitt definiert, dann können nur die Objekte oder die Objektteile grafisch dargestellt werden, deren Punkte innerhalb des festgelegten Raums liegen. Nach der Definition des geometrischen Raumabschnittes ist es notwendig, einen ersten Bezugspunkt innerhalb dieses Raumes zu definieren um die Anfangsposition der darauffolgenden Zeichnung festzulegen. Dies erfolgt durch den Befehl P3D gefolgt von 3 Gleitkommazahlen X, Y und Z, die die entsprechenden Koordinaten des Punktes festlegen, z. B.

P3D 1.0 2.0 0.0

Vom letzten definierten Punkt ausgehend ist es dann durch „S3D“-Befehle möglich, beliebige Punktepaare durch Strecken miteinander zu verbinden. Der S3D-Befehl, hat mit drei reellen Parametern die gleiche Syntax wie der P3D-Befehl und basiert, analog zum S-Befehl in 2D, auf relativen Koordinatenangaben. Abb. 7.1 zeigt ein einfaches Drahtgitterobjekt, das aus der Verbindung von vier Punkten mittels S3D-Befehlen erzeugt wurde:

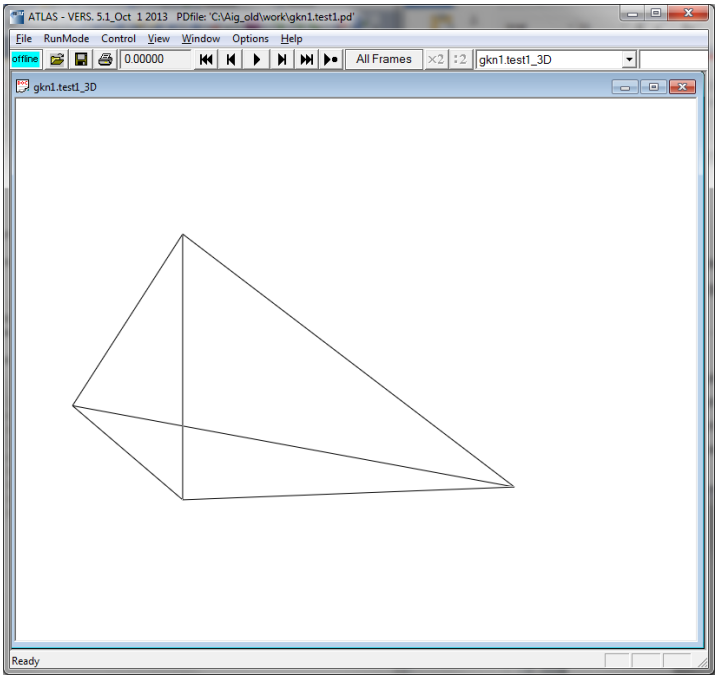


Abb. 7.1 Punkte und Strecken im 3D-Raum

Durch die P3D- uns S3D-Befehle ist es im Prinzip möglich, jede beliebige dreidimensionale Geometrie darzustellen. Zur Vereinfachung der 3D-Geometriedefinition hat es sich allerdings als sinnvoll erwiesen, dreidimensionale Elemente einzuführen, die als elementare Bestandteile komplexer Komponenten verwendet werden können. Dieser Ansatz erweist sich als besonders nützlich für die in AIG genutzte Darstellung der Thermofluidobjekte von ATHLET. Diese bestehen aus Leitungselementen, deren Biegungen und Querschnittsänderungen in 2D durch eine Folge von zusammengesetzten gleichschenkligen Trapezen dargestellt werden. Die Erzeugung einer dreidimensionalen Geometrie kann deswegen einfach durch die Rotation der bildenden Trapeze um ihre Mittellinie erreicht werden. Die dadurch resultierenden Kegelstümpfe bilden somit die Bestandteile jedes beliebigen Thermofluidobjekts in 3D.

Zur geometrischen Darstellung von Kegelstümpfen wurden die folgenden Befehle eingeführt, wobei die Gleitkommazahlen durch Zx und die Ganzzahlen durch Ix wiedergeben werden, Der Anfangspunkt ist der Mittelpunkt der Grundfläche des Kegelstumpfes, der Endpunkt ist der Mittelpunkt der Deckfläche des Kegelstumpfes.

CYLINDER Z1 Z2 Z3 Z4 Z5 I1 I2 I3

Z1: Projektion der Mittellinie des Kegelstumpfes auf der X-Achse

Z2: Projektion der Mittellinie des Kegelstumpfes auf der Y-Achse

Z3: Projektion der Mittellinie des Kegelstumpfes auf der Z-Achse

Z4: Erster Radius des Kegelstumpfes

Z5: Zweiter Radius des Kegelstumpfes

I1: Verknüpfung zum vorhergehenden Kegelstumpf (0 = nein, 1 = ja)

I2: Kontur des Kegelstumpfs (0 = kein, 1 = erster, 2 = zweiter, 3 = beide)

I3: Anzahl der Punkte für die Berechnung des Kegelstumpfes

CYLLYB Z1 Z2 Z3 Z4 Z5 I1 I2 I3

Z1: Länge der Mittellinie des Kegelstumpfes

Z2: Projektion der Mittellinie des Kegelstumpfes auf der Y-Achse

Z3: Winkel zwischen Mittellinie des Kegelstumpfes und XY-Ebene

Z4: Erster Radius des Kegelstumpfes

Z5: Zweiter Radius des Kegelstumpfes

I1: Verknüpfung zum vorhergehenden Kegelstumpf (0 = nein, 1 = ja)

I2: Kontur des Kegelstumpfs (0 = kein, 1 = erster, 2 = zweiter, 3 = beide)

I3: Anzahl der Punkte für die Berechnung des Kegelstumpfes

CYLLAB Z1 Z2 Z3 Z4 Z5 I1 I2 I3

Z1: Länge der Mittellinie des Kegelstumpfes

Z2: Projektion der Mittellinie des Kegelstumpfes auf der Y-Achse

Z3: Winkel zwischen Mittellinie des Kegelstumpfes und XY-Ebene

Z4: Erster Radius des Kegelstumpfes

Z5: Zweiter Radius des Kegelstumpfes

I1: Verknüpfung zum vorhergehenden Kegelstumpf (0 = nein, 1 = ja)

I2: Kontur des Kegelstumpfs (0 = kein, 1 = erster, 2 = zweiter, 3 = beide)

I3: Anzahl der Punkte für die Berechnung des Kegelstumpfes

In Abb. 7.2 werden eine Reihe von Kühlkanälen im Kern durch eine Sequenz von CYLLYB-Befehlen in 3D dargestellt und die lokalen Temperaturwerte farbcodiert angezeigt.

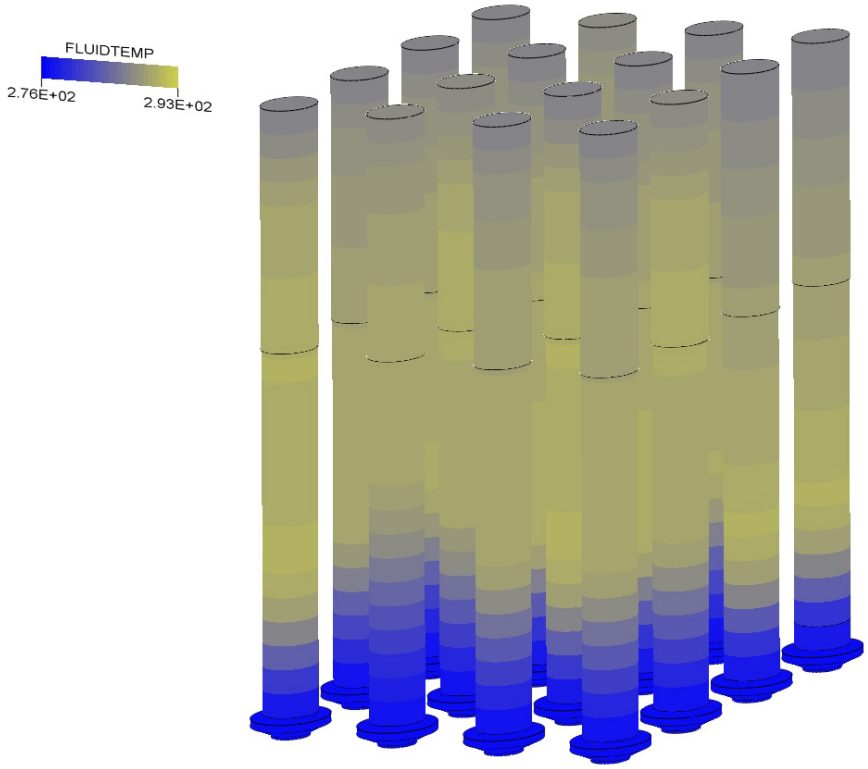


Abb. 7.2 Darstellung von zylindrischen Objekten in 3D

Neben den oben beschriebenen Befehlen zur Darstellung von Kegelstümpfen ist auch ein APG-Befehl zur Erzeugung von Quadern eingeführt worden, wobei der Anfangspunkt eine Ecke des Quaders ist.

CUBOID Z1 Z2 Z3

Z1: Länge der Kante des Quaders in X-Richtung

Z2: Länge der Kante des Quaders in Y-Richtung

Z3: Länge der Kante des Quaders in Z-Richtung

7.1.2 Automatische Erzeugung von 3D-Bildern durch die AIG

Durch die ATHLET Input Graphic ist es möglich, 3D-APG-Bilder der angezeigten Thermofluidobjekte automatisch zu erstellen. In Abb. 7.3 ist das Dialogfenster gezeigt, das vor der Erstellung von APG-Bildern erscheint. Die Bildung dreidimensionaler Bilder erfolgt durch die Wahl der Option 3D.

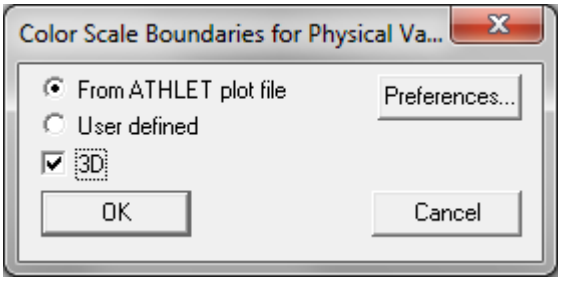


Abb. 7.3 Export von 3D-Bildern aus AIG

In Abb. 7.4 wird gezeigt, wie man einen Winkel zur 3D-Darstellung entlang der Mittellinie eines Thermofluidobjekts, im Beispiel die Volumenausgleichsleitung TFYP10Z200, definieren kann. Im Objektdialog für dieses Objekt wird für den 8. Punkt ein Winkel von 60 Grad vorgegeben. Dabei ist es zu beachten, dass die Winkel zur Zeichnungsebene (XY-Ebene) immer in absoluten Gradangaben angegeben werden müssen.

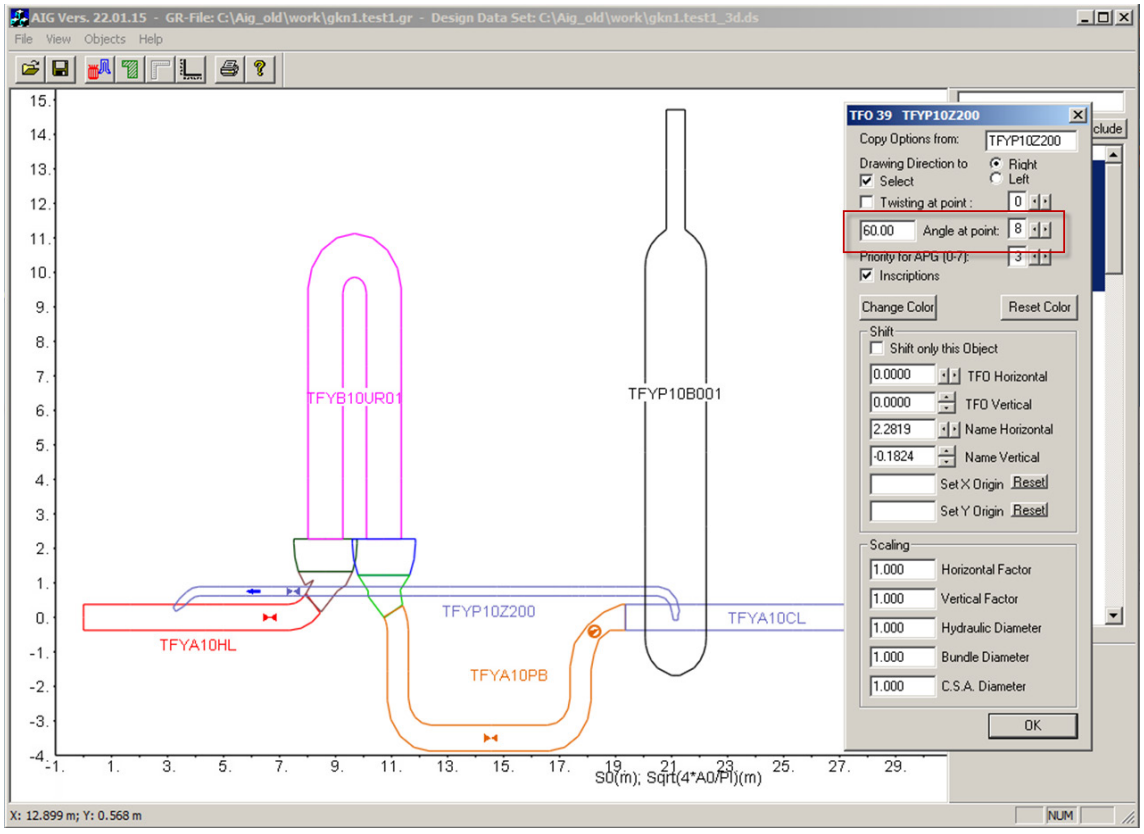


Abb. 7.4 Winkeldefinition für 3D-Darstellung in AIG

Eine Überprüfung der 3D-Darstellung in AIG direkt ist nicht möglich. Es ist nötig, die Darstellung in das ATLAS-APG-Format zu exportieren. In Abb. 7.5 ist dann das Ergebnis, eine 3D-Visualisierung von dynamischen Daten, entsprechend der Vorgaben in

AIG, dargestellt. Im Bild wurde der Blickwinkel (Viewport) so gewählt, dass sich die Objekte nicht überdecken.

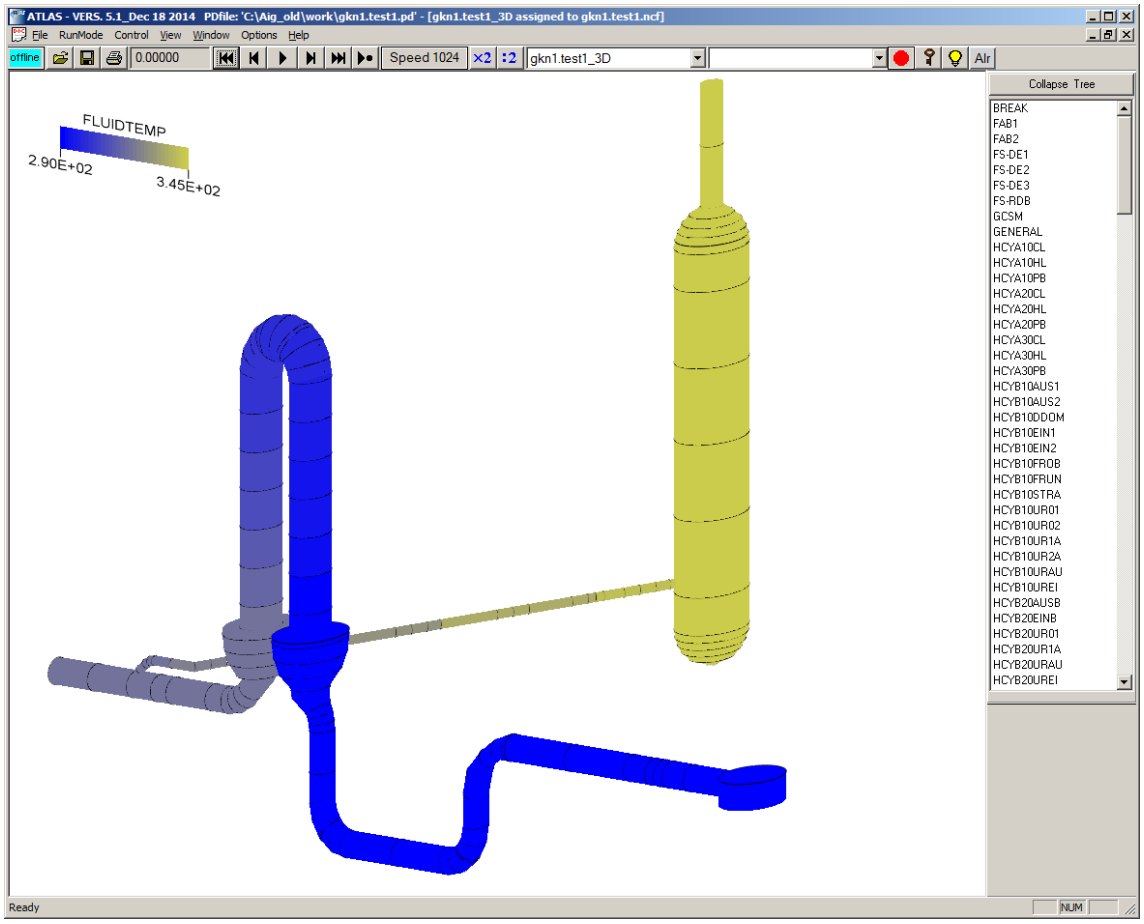


Abb. 7.5 3D-Darstellung einer Nodalisierung in ATLAS

Zur Veränderung des Viewports in 3D-Diagrammen wurde in ATLAS eine Reihe von Tastaturbefehlen implementiert. Durch Betätigung folgender Tasten und Tastenkombinationen kann man das Bild damit beliebig drehen, kippen und verschieben:

- Drehung um die X-Achse in Uhrzeigesinn:X-Taste.
- Drehung um die X-Achse in Gegenuhrzeigesinn: Shift-X-Taste.
- Drehung um die Y-Achse in Uhrzeigesinn: Y-Taste.
- Drehung um die Y-Achse in Gegenuhrzeigesinn: Shift-Y-Taste.
- Drehung um die Z-Achse in Uhrzeigesinn: Z-Taste.
- Drehung um die Z-Achse in Gegenuhrzeigesinn: Shift-Z-Taste.
- Verschiebung nach rechts, links, oben, unten: Pfeil-Tasten.

7.2 Erzeugung von Leittechnikdiagramme mit AGM2APG

7.2.1 Beschreibung des Programms

Das „AGM2APG“-Programm ist eine in der Programmiersprache Python entwickelte Software mit der Aufgabe, die mit dem ATHLET-GCSM-Modeler (AGM) erzeugten GCSM-Signaldiagramme in ein für ATLAS geeignetes Format (APG-Bilddatei) zu konvertieren. Obwohl es sich um ein eigenständiges Programm handelt, ist die Verwendung von AGM2APG ausschließlich in direkter Verbindung mit AGM vorgesehen und es wird deswegen aus der Bedienungsoberfläche dieses Programms gestartet (Abb. 7.6).

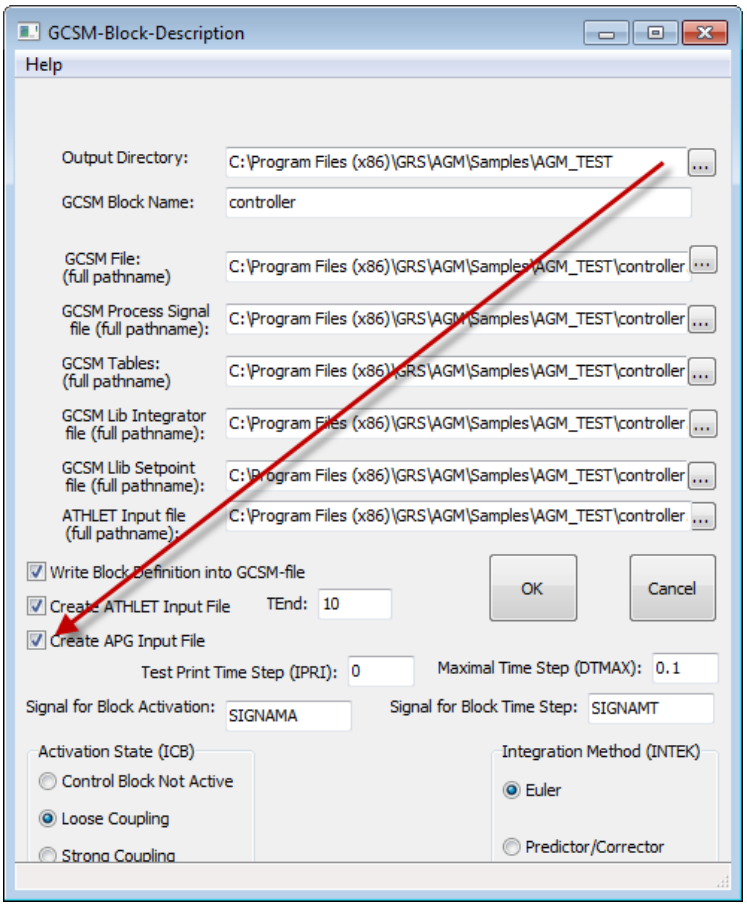


Abb. 7.6 Option zum Export eines Diagramms aus AGM für ATLAS

Beim Starten müssen dem AGM2APG-Programm zwei Parameter übergeben werden. Der erste Parameter entspricht dem Namen eines Files im XML-Format, das von AGM zur Speicherung von Diagrammen optional erzeugt werden kann. Diese Datei enthält

alle Angaben, die zur Beschreibung eines mit AGM erzeugten GCSM-Diagramms erforderlich sind, inklusiv Position und Größe der einzelnen Grafikelemente. Mit diesen Informationen ist durch Einlesen dieser Datei eine Erstellung eines Diagramms mit identischen Eigenschaften zum Original möglich. Die Programmiersprache Python verfügt über eine Programmbibliotheken, die ein sehr komfortables Einlesen und die weitere Verwendung der XML-Daten ermöglichen.

Der zweite Parameter ist der sogenannte Block-Name. Dieser entspricht dem zweiten der vier Keywords, die die GCSM-Signale kennzeichnen und kann vom Benutzer im Exportdialog (Abb. 7.6) vorgegeben werden. In Abb. 7.7 ist die Keywordbaum einer ATHLET-Simulation zu sehen, dessen GCSM-Diagramm mit Hilfe von AGM erstellt wurde. Das zweite Keyword der GCSM-Signale „KBA“ ist in diesem Fall der Block-Name. Der Name des ersten und dritten Keywords ist immer „GCSM“ bzw. „NODE1“. Die Bezeichnungen des vierten Keywords entsprechen den Namen der Signale und werden aus dem durch AGM generierten XML-File entsprechend der verwendeten Elementnamen übernommen.

Ein mit AGM erzeugtes Signaldiagramm kann sehr umfangreich sein. Es kann aus vielen Einzelelementen und auch Diagrammabschnitten gebildet werden, die aus den gleichen Signalstrukturen bestehen und die sich in der Diagrammdarstellung wiederholen können (z. B. für Redundanzen). Um die Übersicht über die Diagrammstruktur auch in solchen Fällen zu gewährleisten, besteht die Möglichkeit, bestimmte Diagrammabschnitte durch einzelne Superblöcke („Submodels“) darzustellen. Mit der Verwendung dieser Submodels ist es dann möglich, Unterdiagramme zu verwalten. Deswegen besteht eine mit AGM erzeugte Signalstruktur aus einem Hauptdiagramm und eventuell aus einer Reihe zusätzlicher Unterdiagramme, die vom Hauptdiagramm und voneinander hierarchisch abhängig sein können.

In Rahmen der Simulatoranwendung ist es wichtig, dass die Funktionalität in Verbindung mit der beschriebenen Diagrammabhängigkeit erhalten bleibt. Deswegen wird bei der Konversion von AGM-Diagrammen in APG-Files nicht nur aus dem Hauptdiagramm ein entsprechendes APG-File mit der gleichen Signalstruktur erzeugt. Es werden auch für die einzelnen Unterdiagramme zusätzliche APG-Files angelegt. Die Diagrammstruktur bleibt somit erhalten und die erzeugten APG-Files entsprechen dann genau in AGM definierten Organisation in Submodels.

Die Ausgabedateien werden von AGM2APG daher folgendermaßen organisiert. Es wird zuerst ein Verzeichnis und eine Datei mit dem gleichen Namensstamm des ersten Parameters aber mit der Erweiterung „_all.apg“ angelegt. Beispielsweise wird mit Eingabefile „KBA.xprt“, dann wird ein Verzeichnis „KBA“ und eine Ausgabedatei mit dem Namen „KBA_all.apg“ erzeugt. Diese beinhaltet nur Befehle (INCLUDE-Anweisungen), die in ATLAS zum Laden aller erzeugten APG-Files erforderlich sind. Die APG-Files selbst, welche die Signaldiagramme reproduzieren, werden ebenfalls in dem erzeugten Verzeichnis gespeichert.

7.2.2 Verwendung der Bilder in ATLAS

Wenn in ATLAS die Datei mit der Erweiterung „_all.apg“ über die Option „Add Picture“ eingelesen wird, dann werden alle dazugehörigen Bilder geladen und in die Picture-Liste (linke Combobox in Abb. 7.7) aufgenommen. Sie erscheinen aber mit Ausnahme des Hauptdiagramms nicht auf dem Bildschirm.

Abb. 7.7 zeigt einen ATLAS-Lauf, in dem ein durch AGM2APG generiertes, aus zwei APG-Files bestehendes, Bild-Set, geladen wurde. Das obere Fenster stellt das Hauptdiagramm der GCSM-Signale dar. Im unteren Fenster wird das vom Hauptdiagramm hierarchisch abhängige Submodel visualisiert. Dieses stellt die Struktur des unter der Bezeichnung „FF-DYN2“ gekennzeichneten Signalblocks dar.

Nach dem Laden des Bild-Sets erscheint nur das erste Fenster mit dem Hauptdiagramm. Das zweite Fenster kann entweder aus der linken Combobox, die die gesamte Liste aller geladenen APG-Files beinhaltet oder einfach durch Mausklick auf den „FF-DYN2“ Block aktiviert werden.

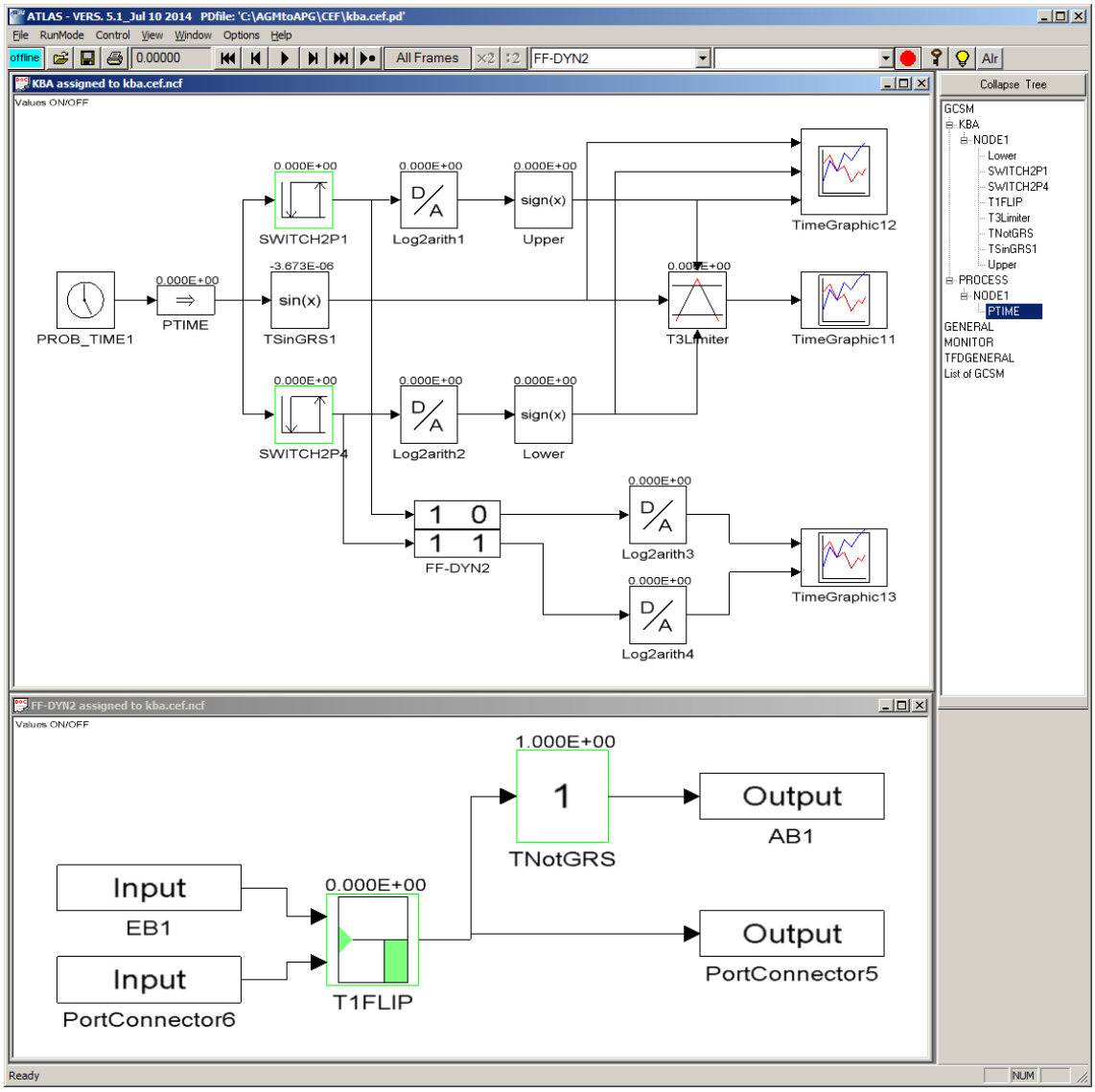


Abb. 7.7 GCSM-Diagramme in ATLAS generiert durch AGM2APG

Alle Elementsymbole in Abb. 7.7 stellen entweder einzelne GCSM-Signalblöcke oder systeminterne Blöcke von AGM dar. Wenn die Maus auf die GCSM- Signalblöcke geführt wird, verändert sich der Zeiger vom Pfeil zum Zeigefinger. Das ist ein Hinweis dafür, dass durch Doppelklick auf den Block ein Zeitdiagramm geöffnet werden kann, in dem der Wert des Signals als Funktion der Zeit dargestellt wird. Subworkspaces können hingegen aus der Veränderung des Mauszeigers zum horizontalen Doppelpfeil erkannt werden. Durch einen Mausklick auf die Option „Values ON/OFF“ in der linken oberen Fensterecke der Signaldiagramme, die Anzeige des aktuellen Ausgangswerts über den Elementsymbolen ein bzw. ausgeschaltet werden kann.

Es können mit AGM sehr komplexe Signaldiagramme erstellt werden, die umfangsreichen Verschachtelungen aufweisen können. Die Kette der Abhängigkeiten in den baumartigen Strukturen kann beliebig lang sein, so dass die einzelnen Diagramme weitere Diagramme enthalten und diese wiederum andere. Programmintern ist dieses Problem durch die Verwendung der Rekursion gelöst worden. Die Routine, die aus den einzelnen Diagrammen APG-Files erzeugt, ruft sich solange selbst auf bis die Signaldiagramme erreicht sind, die keine Submodels mehr aufweisen.

7.3 Shapiro Diagramm

Zur Einschätzung des Brand- und Explosionsrisikos in kerntechnischen Anlagen stellt das Shapiro-Diagramm Stoffkonzentrationen in einem Dreistoffdiagramm dar. Die in solchen Diagrammen dargestellten Verbrennungs- und Detonationslinien des Luft-Dampf-Wasserstoff-Gemisches markieren die Grenzen eines Sicherheitsbereichs, dessen Überschreitung eine erhebliche Risikorelevanz für die Anlage aufweisen kann.

Eine Funktion zur interaktiven Erzeugung eines Shapiro-Diagramms in ATLAS wurde deshalb ergänzt. Ausgehend von vorhandenen, manuell im ATLAS Bildeditor erstellten Diagrammen wurde die Bilderstellung automatisiert. In der aktuellen ATLAS-Version können mit der neuen Funktion „Shapiro“ für MELCOR, ASTEC und COCOSYS entsprechende Bilder (APG-Files) interaktiv, automatisch erzeugt werden, die für eine oder mehrere Zonen im Sicherheitsbehälter Dreistoffdiagramme darstellen. Die Generierung dieser Bilder erfordert daher nur geringen Aufwand und keine Kenntnisse der Bildbeschreibungssprache APG von ATLAS.

Die Shapiro-Funktion lässt sich im Menü „Options“ der Menüleiste (Abb. 7.8) von ATLAS aufrufen.

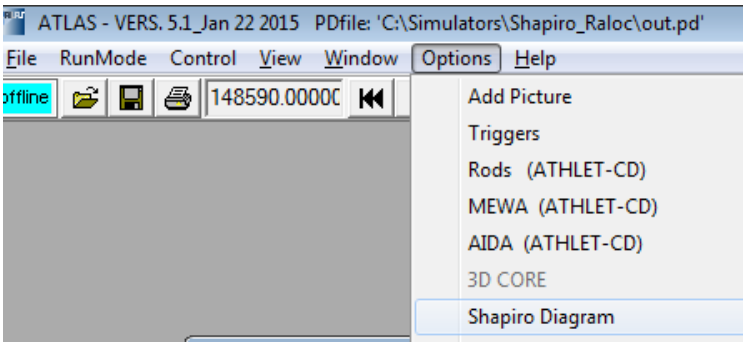


Abb. 7.8 Shapiro Diagramm in der ATLAS-Menüleiste

Zur Verfügbarkeit der Funktion müssen gewisse Voraussetzungen erfüllt sein, die sich von Rechencode zu Rechencode unterscheiden. Für alle Programme gilt aber, dass es möglich sei muss, eine Liste von Räumen aufzustellen, in denen die Berechnung der Zusammensetzung des Luft-Dampf-Wasserstoff-Gemischs aus den Daten des Rechenlaufs erfolgen kann. Die Kriterien, die für die Erstellung einer Liste von Räumen angewendet werden, sind im Folgenden für die einzelnen Rechencodes beschrieben:

Für COCOSYS müssen das erste Keyword „RALOC“ und das vierte Keyword „ZCVOLP“ heißen. Es werden dann aus dem zweiten Keyword diejenigen Zonen in die Liste aufgenommen, deren Endungen „_GS“ oder „_GM“ lauten.

Für ASTEC muss die Keywords-Kombination „CONTAINM … THER H2“ oder, „CONTAINM … THER XH2“ sein. Dann ist das zweite Keyword die Bezeichnung einer Zone und wird in die Liste aufgenommen.

Für MELCOR sucht das Programm nach Keywords-Kombinationen, deren erste drei Bezeichnungen „CVH“, “X“ und „3“ lauten. Ist in der Keywords-Liste eine solche Zeile vorhanden, dann ist das vierte Keyword die Bezeichnung einer Zone, die die oben beschriebenen Kriterien erfüllt. Sie wird dann in die Liste der Räume aufgenommen.

Kann durch die Erfüllung der oben beschriebenen Kriterien eine Liste von Räumen gebildet werden, dann wird der Menüpunkt „Shapiro Diagram“ aktiviert und somit kann der Aufruf der Dreistoff-Funktion erfolgen.

Nach der Auswahl der Funktion wird zuerst die Liste der Räume angezeigt (Abb. 7.9). Aus ihr kann der Anwender eine oder mehrere Zonen auswählen, deren Gemisch-Zusammensetzung in einem einzelnen Shapiro-Diagramm grafisch dargestellt werden soll.

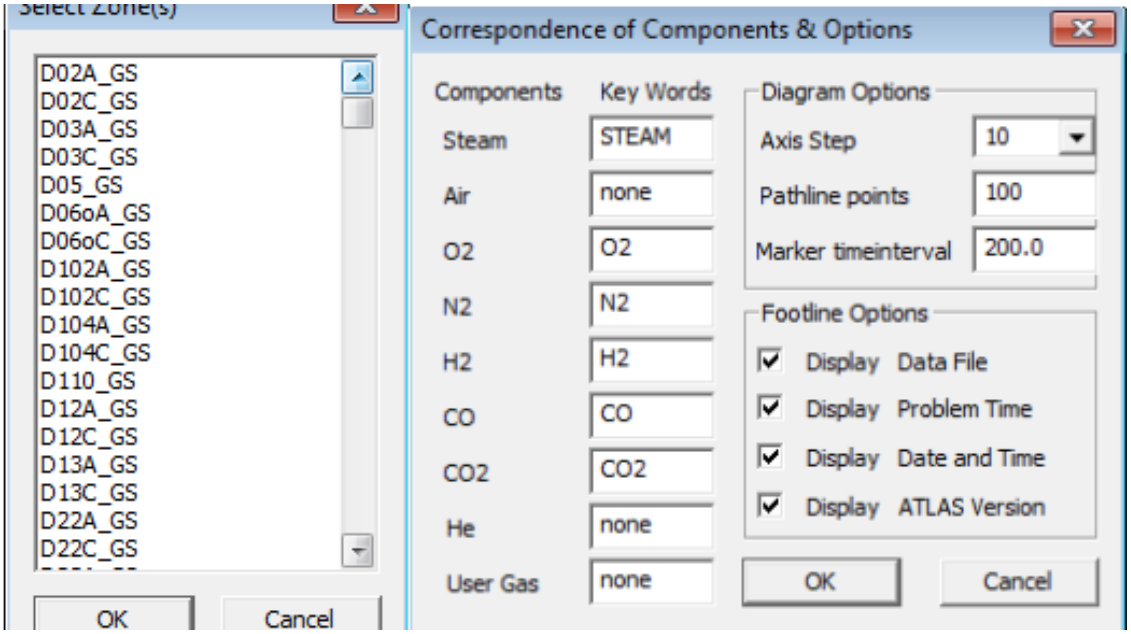


Abb. 7.9 Raumauswahl und Optionen für Shapiro Diagramm

Nach Auswahl der Räume durch „OK“ erscheint der Shapiro-Optionendialog. Auf der linken Dialogseite werden die in der Keywords-Liste gefundenen Bezeichnungen der verschiedenen wichtigen Stoffe des Gemisches aufgelistet. Sollten diese Bezeichnungen für den betreffenden Lauf nicht zutreffen, kann der Anwender sie durch andere Keywords ersetzen. Letztere müssen den in der Keyword-Liste des Rechenlaufes vorhandenen Stoffen entsprechen.

Auf der rechten Dialogseite werden weitere Optionen zur Erstellung des Shapiro-Diagramms angeboten. Der Anwender kann damit die grafische Gestaltung des Diagramms beeinflussen.

- Die Option **„Axis Step“** bestimmt die Maschenabstände des Gitternetzes in dem Phasendreieck.
- Die Option **„Pathline points“** bestimmt die maximale Anzahl der Punkte, die im Diagramm zur Darstellung des Verlaufs eines Stoffgemischs für einen Raum verwendet wird.
- Die Option **„Marker timeintervall**“ gibt in Sekunden an, nach welchen Zeitabständen die Markierung des Gemisch-Pfades durch kleine quadratische Symbole erfolgen soll.

Schließlich, legen die letzten vier Optionen des Dialogs die Gestaltung der Fußzeile im Bild fest.

Abb. 7.10 zeigt ein Shapiro Diagramm mit dem Verlauf der Gasgemische für 4 Räume entsprechend einer mit COCOSYS durchgeführten Rechnung. Im Diagramm ist erkennbar, dass bis zu der angezeigten Problemzeit für die verschiedene Zonen die Verbrennungsgrenzen des Wasserstoffes und des Kohlenmonoxids überschritten worden sind. Dies ist immer dann der Fall, wenn Punkte der Verlaufskurven in den entsprechend gekennzeichneten Bereichen (Gelb, Orange) liegen. Die Verlaufslinien der Gasgemische werden nur dann vollständig angezeigt, wenn ATLAS die Daten der Simulation kontinuierlich wiedergibt. Bei „Jump to Time“ wird nur die Konzentration zum aktuellen Zeitpunkt dargestellt. Die Berechnung der Punkte für die Gasgemische ist im Anhang 2 näher erläutert.

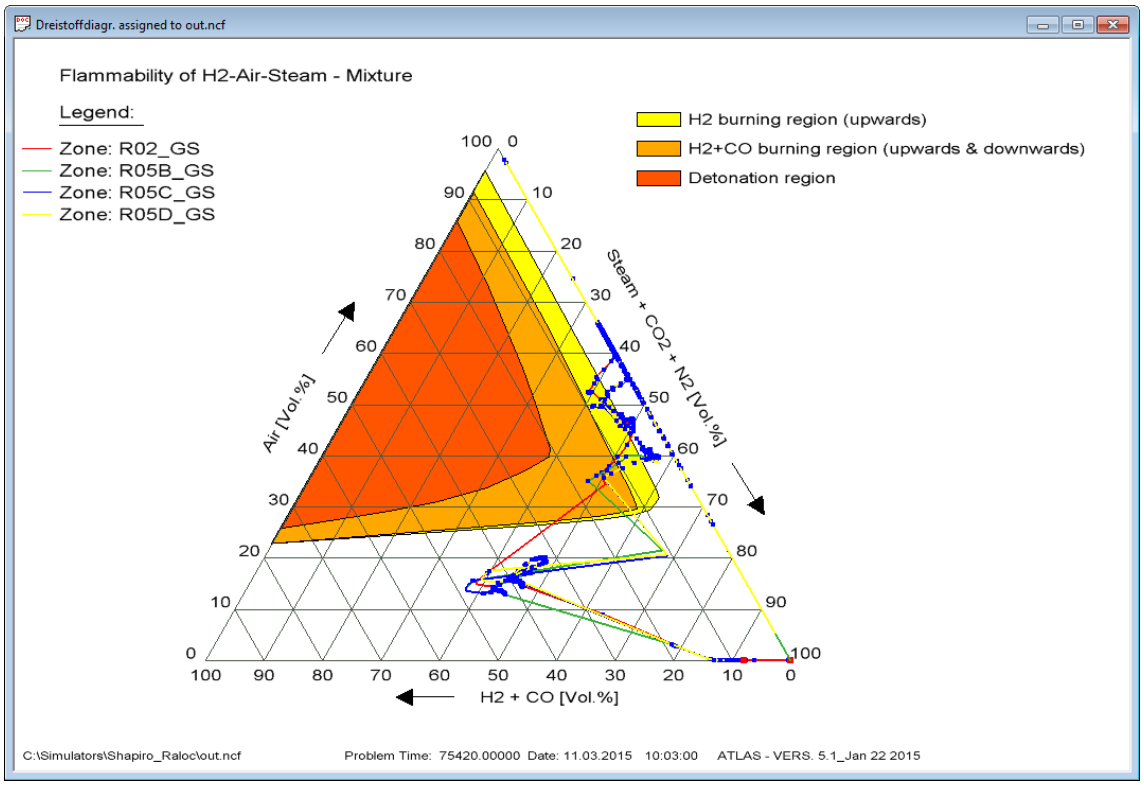


Abb. 7.10 Beispiel für ein Shapiro Diagramm

Im Diagramm werden an der linken Dreieckseite die Luft, an der rechten die sogenannten Inertgase und unten die entzündbaren und detonationsfähigen Gase aufgetragen. Falls weitere Gase wie Helium oder andere, die vom Anwender unter der Bezeichnung „User Gas“ angegeben werden können, vorhanden sind, werden diese zu den Inertgasen addiert.

Wie ebenfalls aus Abb. 7.10 ersichtlich, können durch die Wahl der oben erwähnten Fußzeile-Optionen am unteren Rande des Dreistoffdiagramms der Dateiname der Simulationsdaten, die aktuelle Problemzeit, das Datum und die Uhrzeit sowie die ATLAS-Version eingeblendet werden.

7.4 Vergleich von Simulationsdaten

7.4.1 Allgemein

Das Hinzufügen („Laden“) zusätzlicher Simulationsdateien im NETCDF-Format („NC-Files“) ermöglicht es dem ATLAS-Anwender, Vergleiche zwischen Ergebnissen zweier oder mehrerer Rechenläufe durchzuführen. Verglichen können Rechnungsergebnisse sowohl aus ATHLET, als auch aus anderen durch ATLAS unterstützten Rechnungsprogrammen wie COCOSYS, ASTEC und MELCOR. Es ist nicht möglich, die Ergebnisdateien im Originalformat der Programme direkt zu laden. Falls die NETCDF-Dateien nicht bereits während der Simulation erzeugt wurden, kann man Originaldateien mit dem ATLAS „Data Viewer and Converter“ (Abb. 7.11) in das benötigte Format umwandeln. Der Konverter kann aus der ATLAS Programmgruppe gestartet werden.

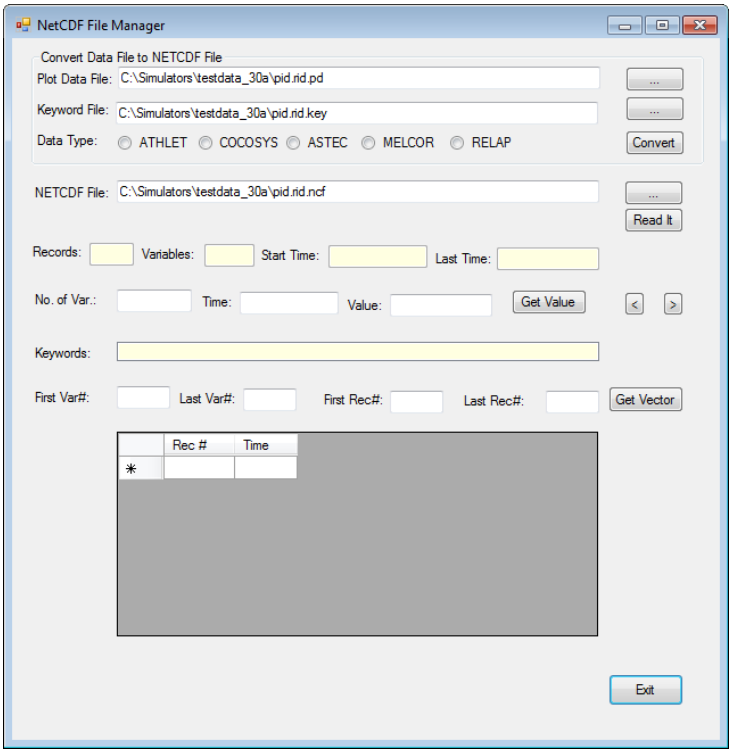


Abb. 7.11 Grafische Oberfläche zur Erzeugung von NetCDF-Dateien

Nach dem Starten von ATLAS ist das Laden zusätzlicher Simulationsdateien mit „Add NC File“ unter dem Reiter „File“ der Hauptbedienleiste möglich.

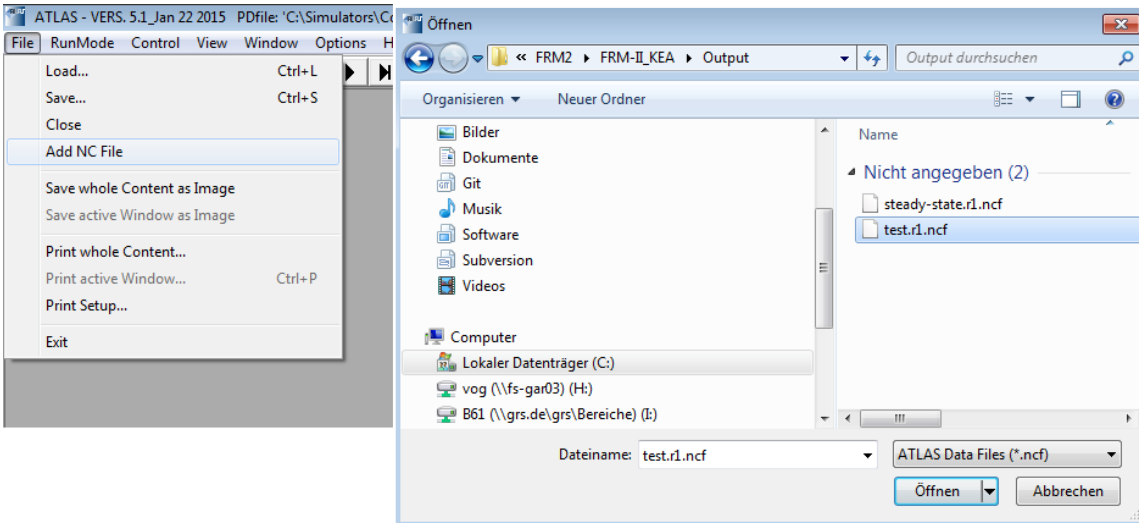


Abb. 7.12 Zusätzliche Simulationsdateien hinzufügen

Es können außer der beim Programmstart verwendeten Datei bis zu 9 weitere NC-Files verwendet werden. Nach dem Einlesen einer neuen Datei wird die Keyword-Liste reorganisiert. Sie wird durch das Einfügen der Keywords für die neuen Daten erweitert.

Keywords, die gleiche Bezeichnungen haben wie die schon vorhandenen, werden unter den gleichen Namen im Baum integriert und der Name des vierten Keywords wird durch einen zusätzlichen Indikarot zwischen „_1“ und „_9“ ergänzt. Aus diesen Angaben ist es dann möglich, die Herkunft der Größen zu erkennen. Demzufolge stammt z. B. eine Temperaturangabe, die in der Keyword-Liste unter der Bezeichnung TLS_2 erscheint, aus dem zweiten dazu geladenen NC-File (siehe Abb. 7.13). Mit diesem Vorgehen ist dann beispielsweise möglich, Simulationsläufe mit identischen Eingabedaten und Keywords zu vergleichen, die mit unterschiedlichen Programmversionen der Rechencodes durchgeführt wurden.

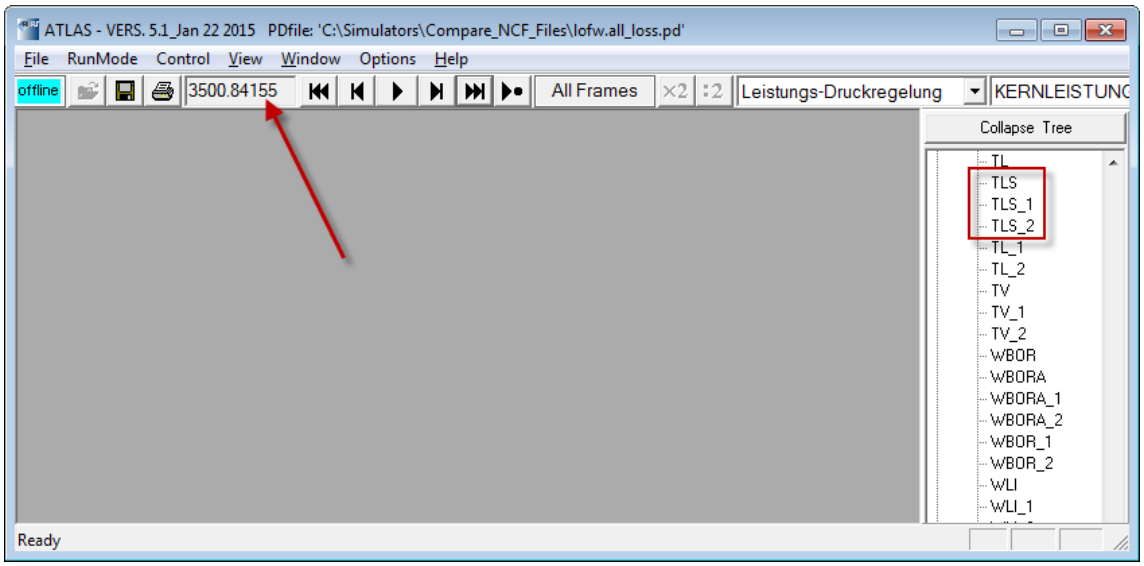


Abb. 7.13 Zeiten und Keywords bei mehreren Simulationsdateien

Neben der Keyword-Liste, wird nach dem Einlesen zusätzlicher NC-Files auch der Zeitvektor der Rechnung reorganisiert, indem, falls vorhanden, neue Zeitpunkte aus den nachgeladenen NC-Files hinzugefügt werden. Der Zeitvektor des Rechenverlaufs resultiert somit aus der Summe der Zeitvektoren aller geladenen NC-Files. Beim Laufen der Simulation erscheinen somit alle Zeiten aller geladenen Rechnungen in der Anzeige (Abb. 7.13). Dabei ist es zu beachten, dass dennoch die Daten aus den einzelnen NC-Files ihre eigenen Zeitvektoren beibehalten. So wird der Verlauf von Zeitdiagrammen, die aus einer bestimmten Rechnung stammen nicht durch das Laden von Daten aus anderen Rechnungen beeinflusst, denn jedes Zeitdiagramm stützt sich weiter nur auf die im eigenen NC-File vorhandenen Zeiten.

7.4.2 Zeitdiagramme aus mehreren NC-Files

Die Möglichkeit Zeitdiagramme mit Daten aus mehreren NC-Files verwenden zu können erleichtert den Vergleich des Verlaufs von physikalischen Größen aus unterschiedlichen Rechnungen erheblich.

Das Zeitdiagramm einer physikalischen Größe aus einem beliebigen NC-File lässt sich wie gewöhnlich durch Doppelklick aus der Keywords-Liste abrufen. Da für den Aufbau eines einzelnen Zeitdiagramms immer der zugeordnete Zeitvektor verwendet wird, können die dargestellten Kurven unterschiedliche Anfangs- und Endzeiten haben. Die Länge der Zeitachse aller Zeitdiagramme wird aber aus dem Gesamtzeitvektor, also der Summe aller einzelnen Zeitpunkte, bestimmt und ist deswegen für alle Diagramme gleich. Es kann daher Zeitbereiche geben, wo für die jeweiligen Keywords keine Daten vorhanden sind. Die Darstellung der betroffenen Zeitkurven (Abb. 7.14) wird in diesen Zeitbereichen unterbrochen, d. h. es werden keine Kurvenwerte gezeichnet.

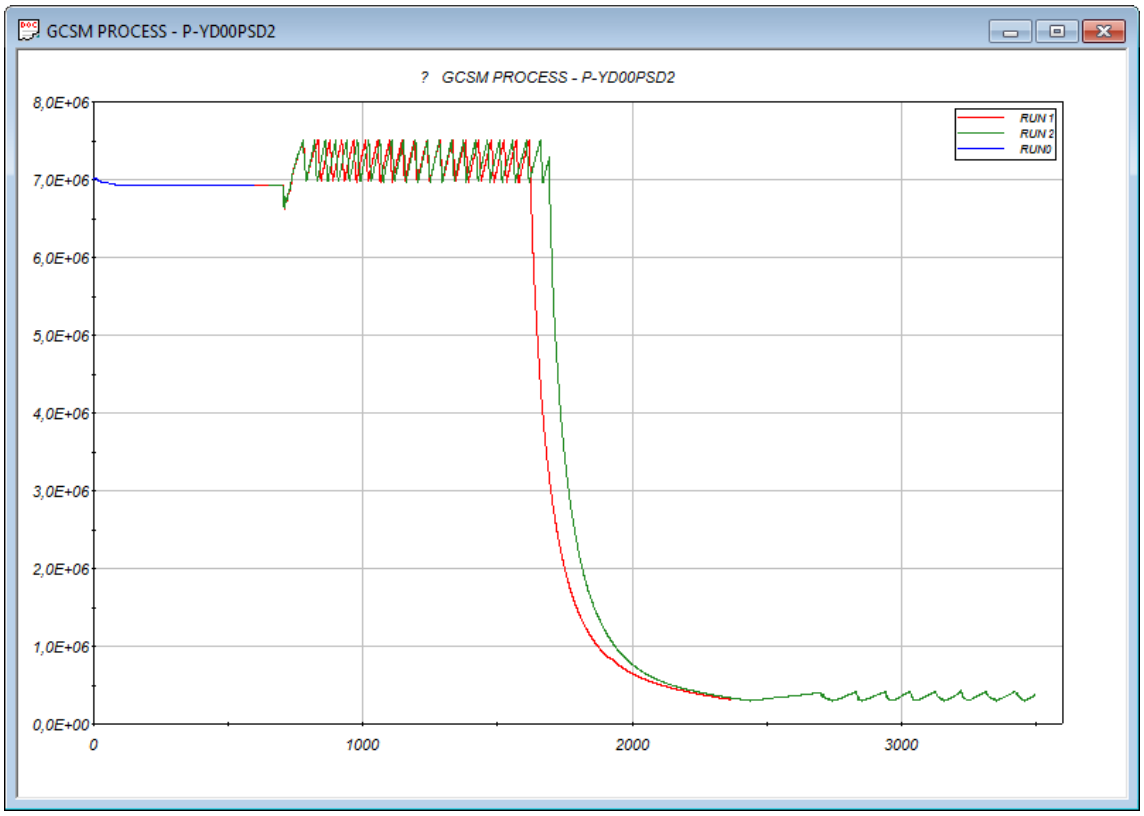


Abb. 7.14 Zeitkurven von Größen aus verschiedenen Rechenläufen

Wie aus dem Bild ersichtlich, endet „RUN0“ bei ca. 600s, während „RUN1“ und „RUN2“ bei diesem Zeitpunkt beginnen. RUN2 enthält weitere Zeiten jenseits des Endes von RUN1 bei ca. 2400s.

7.4.3 Zuordnung der APG-Bilder zu den NC-Files

Beim Start von ATLAS werden die Prozessbilder immer mit den Daten, die im Startmenü von ATLAS definiert sind, verknüpft. Sind während einer Simulation weiter NC-Files geladen worden, werden danach hinzugefügte Prozessbilder per Default dem letzten geladenen NC-File zugeordnet. Eine Verknüpfung eines Bild mit Daten aus anderen NC-Files kann durch die Option „Assign Picture to NC-File“ aus dem Kontextmenü (Abb. 7.15) erreicht werden, das für jedes Bild durch einen rechten Mausklick abrufbar ist.

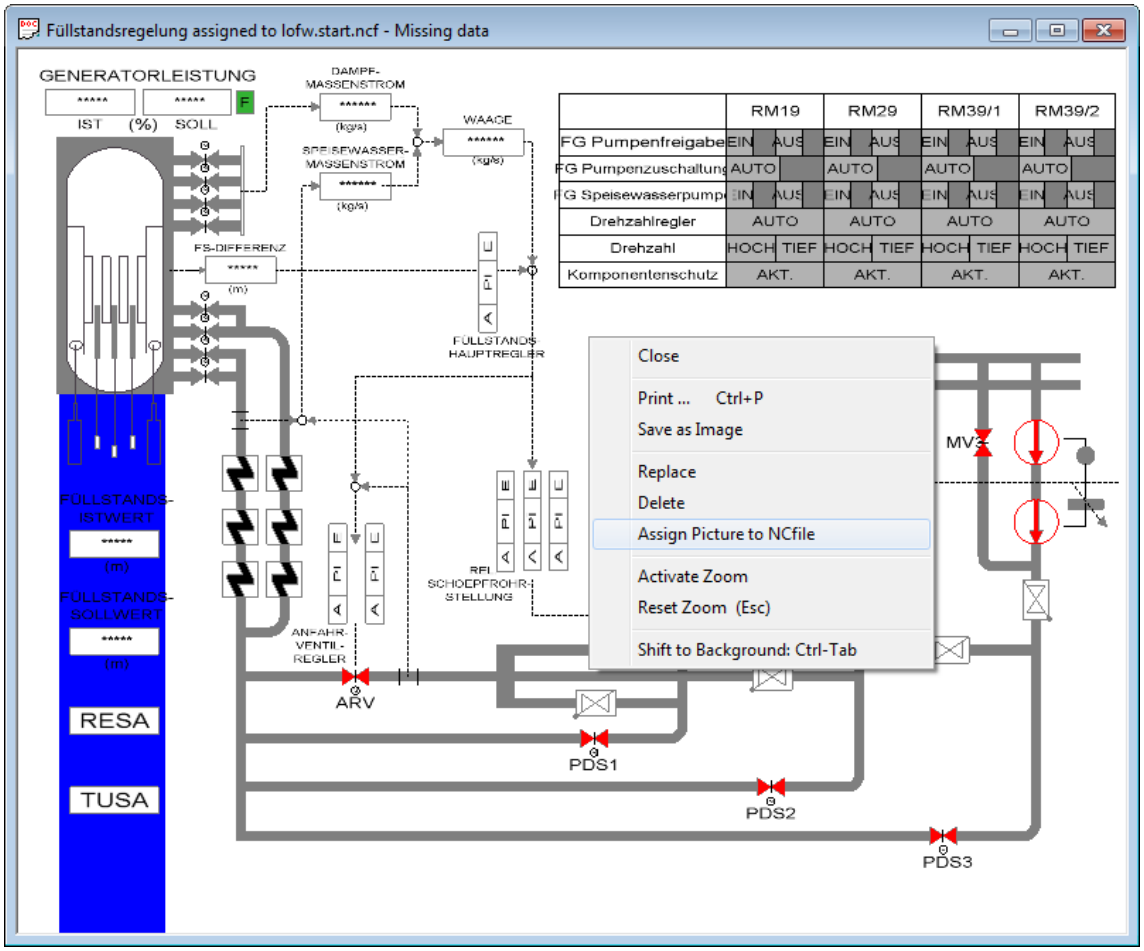


Abb. 7.15 Zuordnung von Prozessbildern zu Rechenläufen mit undefinierten Daten

Im Fensterrahmen des Bildes wird der Name des aktuell zugeordneten NC-File angezeigt. Falls zum aktuellen Zeitpunkt dort keine Daten vorliegen, wird darauf mit „Missing data“ hingewiesen. Alphanumerische Werte werden dann als „*****“ dargestellt. Andere grafische Anzeigen, wie Farbe oder Geometrie von Grafikelementen, benutzen als Ersatzwert die Zahl „-1-0E25“, also eine sehr große negative Zahl. Bei einer Auswahl einer Datei (Abb. 7.16), für die für den aktuellen Zeitpunkt Daten vorliegen, ergibt sich dann wieder eine korrekte Darstellung entsprechend der Daten.

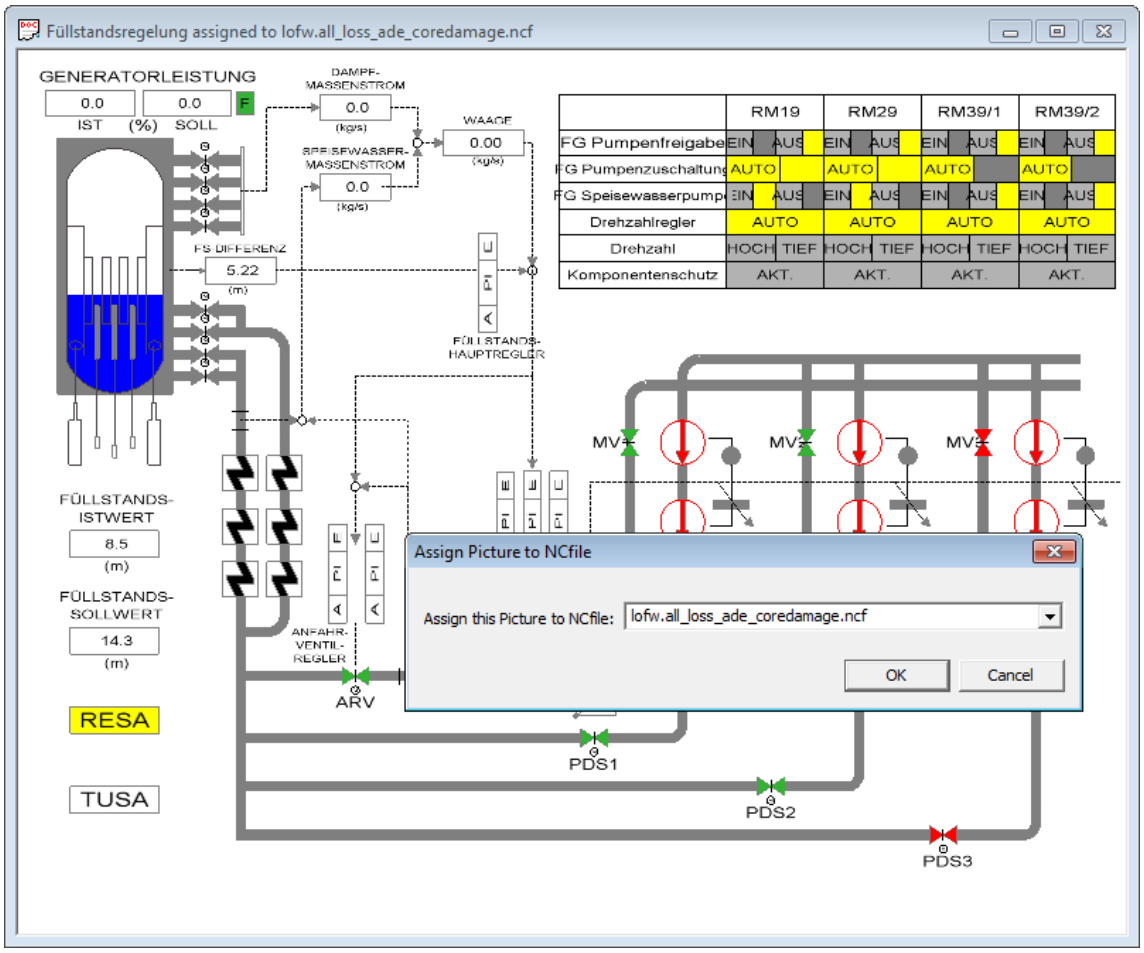


Abb. 7.16 Zuordnung von Prozessbildern zu Rechenläufen mit definierten Daten

7.4.4 Speicherung von ATLAS-Simulationen mit mehreren NC-Files

Werden im Laufe einer ATLAS-Analyse mehrere NC-Files geladen, so lässt sich der Lauf zu einem späteren Zeitpunkt unter den gleichen Bedingungen wiederholen, indem man durch die Funktion „Save…“ im File-Menü die aktuelle Konfiguration in einem sogenannten „Graphic Definition File“ mit der Erweiterung „*.apt“ speichert. In dieser

APT-Datei werden Angaben über die angezeigte Zeitdiagramme und Bilder und deren Lage gemacht. Ab der ATLAS-Version 5.1 werden unter dem neu eingeführten Schlüsselwort „[added_ncfiles]“ auch die Namen der geladenen NC-Files gespeichert. Durch das erneute Einlesen der APT-Datei werden dann auch die aufgelisteten NC-Files nacheinander geladen und damit die Kurven in den Zeitdiagrammen korrekt wiederhergestellt. Die Bilder bedürfen einer erneuten manuellen Zuordnung und sind anfangs wie oben beschrieben zugeordnet.

7.5 ATLAS 5.1 Release

Während der Vorhabenslaufzeit wurden in der freigegebenen Version 5.0 von ATLAS verschiedene Fehler im Programm beseitigt. Die oben beschrieben neuen Funktionen wurden integriert. Die Programmdokumentation, die „User Manuals“ für ATLAS /VOG 14A/ und APG /VOG 14B/ wurden erweitert und verbessert.

Vor der Freigabe der neuen Version wurden umfangreiche Tests der Programme durch die Entwickler und interne Anwender durchgeführt. Die gefundenen Probleme in den neu entwickelten Funktionen wurden beseitigt. Alle Änderungen wurden in einer neuen ATLAS Release Version 5.1 zusammengefasst. Die Dokumentation, Anwendungsbeispiele und alle notwendigen Programmmodule wurden in ein Installationsprogramm für Windows zusammengefasst. Die neue Programmversion kann auf den Windows Versionen XP/SP3, 7 und 8 genutzt werden

Das Installationsprogramm für ATLAS 5.1 wurde auf der ATLAS User‘ Area (Abb. 7.17) allen externen Nutzern mit gültigen Lizenzverträgen (ca. 30) zum Download zur Verfügung gestellt.

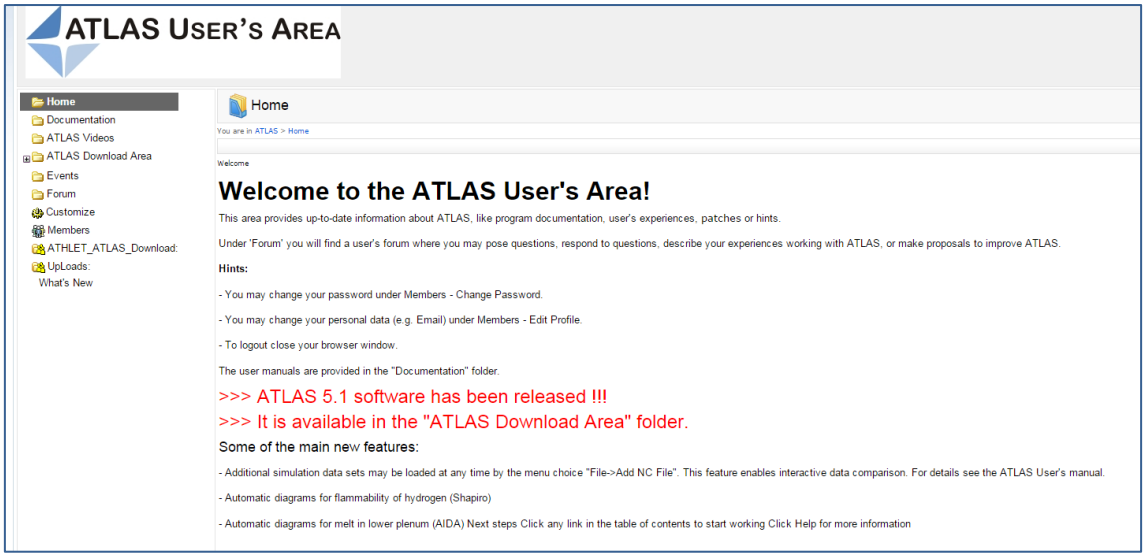


Abb. 7.17 ATLAS User's Area

8 Qualitätssicherung und Qualitätsmanagement

Die GRS ist nach der prozessorientierten DIN ISO 9001:2000 Regel zertifiziert. Der Geltungsbereich dieser Zertifizierung betrifft Forschung und Entwicklung, Sicherheitsbewertung und Projektträgerschaft auf dem Gebiet der Anlagen- und Reaktorsicherheit, Entsorgung, Endlagerung sowie Umweltschutz.

Die GRS-Geschäftsprozesse sowie das Qualitätsmanagementprogramm mit der Verbindlichkeitserklärung der Geschäftsführung sind auf der GRS-Homepage /GRS 15/ zu finden.

Die in diesem Projekt durchgeführten Entwicklungsschritte wurden mit Hilfe des Projektmanagementsystems Trac /TRA 15/ geplant und dokumentiert. Dieses lizenzfrei einsetzbare System vereint verschiedenste Werkzeuge zur Verwaltung von Projekten aller Art und stellt diese den Entwicklern als webbasierte Anwendung zur Verfügung. Den zentralen Bestandteil bildet hierbei das Wiki-System, welches sich durch seine einfache Syntax hervorragend zur schnellen Dokumentation von Entwicklungsschritten oder Zwischenergebnissen eignet. Wie für Wiki-Systeme typisch, können Beiträge und Änderungen direkt im Browser als schlicht formatierte Texteingaben eingegeben werden. Beim Absenden werden diese in fertige HTML-Seiten umgewandelt und stehen anderen sofort zur Verfügung. Die Möglichkeiten bei der Umwandlung können durch verschiedene Plugins erweitert werden und erlauben so z. B. auch das einfache Erstellen komplexer UML-Diagramme.
Sowohl erkannte Programmfehler als auch die Liste geplanter Erweiterungen werden im integrierten Bugtrackingsystem als Tickets erfasst und verwaltet. Die Bearbeitung eines Problems und die zu dessen Lösung durchgeführten Maßnahmen können direkt im jeweiligen Ticket dokumentiert werden und sorgen durch eine Verlinkung zur entsprechenden Revision in der Quellcode-Versionsverwaltung mit Subversion (SVN) für eine sehr gute Nachvollziehbarkeit von Quellcodeänderungen.
Durch das in Trac vorgesehene Zusammenfassen von Tickets zu Meilensteinen wird der zeitliche Verlauf des Projekts in Form eines Projektplans organisiert. Damit können ungelöste Probleme im Auge behalten und die zum Erreichen eines Meilensteins, wie die Freigabe der nächsten Release-Version, notwendigen Arbeiten geplant werden.

9 Zusammenfassung und Ausblick

Das übergeordnete Ziel der Arbeiten in RS1199 war die Entwicklung generell einsetzbarer Werkzeuge, die mit interaktiven, grafischen Methoden das Prä- und Postprozessing von Simulationscodes der Reaktorsicherheit unterstützen. Ein besonderer Schwerpunkt waren die durch den Systemcode ATHLET gestellten Anforderungen.

Im Ergebnis steht ein CAD-System zur Verfügung, das die Vorgabe von 3D-Geometrie und Gitterdaten (Diskretisierung) ermöglicht. Es basiert auf Erweiterungen der Open Source-Entwicklung FreeCAD. Es bietet Schnittstellen, die zur Erzeugung der Eingabedaten der Codes und zum Datenexport für das Visualisierungswerkzeug genutzt werden können.

Das Gesamtsystem wurde mit verschiedenen Anwendungen umfangreich getestet. Im CAD-System wurden die Hauptkühlmittelleitungen und der Druckbehälter eines DWR modelliert und die entsprechende 3D-Konstruktion erstellt. Die interaktive Erstellung und Modifikation der Rechengitter auf diesen Geometrien ist mit den entwickelten Erweiterungen mit relativ geringem Aufwand möglich. Die Daten wurden als Grundlage für die Störfallsimulation mit ATHLET verwendet. Der Export der Daten aus der Simulation und dem CAD-System konnte mit den neuen Konvertierungswerkzeugen für das EXODUS II-Format durchgeführt werden. Die konvertierten Daten sind die Grundlage für die 3D-Visualisierung in ParaView. Die enthaltene Diskretisierungsinformation erlaubt eine gute Navigation in den lokalen Daten und die einfache Erstellung von Ansichten, die die Analyse optimal unterstützen.

Zur Modellierung thermohydraulischer und leittechnischer Systeme wurde ein generischer, grafischer Netzwerkeditor adaptiert und erweitert. Dieser dient dazu, Netzwerke mit komplexer Topologie aus geeigneten Bausteinen und Elementen nachzubilden. Die Bibliotheken der Basiselemente können entsprechend der Modellanforderungen erweitert werden. Die Bausteine können interaktiv in einer Zeichenebene zu einem Gesamtmodell verbunden werden. Nach Vorgabe der Parameter für die Elemente können die Eingabedaten im codespezifischen Format erzeugt werden. Bei Bedarf kann eine Modellierung mit einer internen Simulation im Netzwerkeditor ergänzt werden.

Der Netzwerkeditor wurde zur Modellierung von Leittechnik- und Thermofluidsystemen in ATHLET ausgebaut und angewendet. Das Werkzeug zur Leittechnikmodellierung AGM hat im Vorhaben einen Entwicklungsstand erreicht, der bereits für die die prakti-

sche Anwendung zur Erstellung komplexer Systeme qualifiziert ist. Die Modellierung von Fluidsystemen mit ATM unterstützt hingegen nur einen Teil des Modellumfangs in ATHLET und benötigt außerdem weitere Verbesserungen in der Benutzeroberfläche. Dennoch konnte mit dem erstellten Prototyp die prinzipielle Eignung des Netzwerkeditors auch für diesen Bereich gezeigt werden.

Zur Auswertung der Simulationsergebnisse und ihre visuelle Darstellung im lokalen Zusammenhang mit der 3D-Geometrie und dem Simulationsgitter wurde das Open-Source-Produkt „ParaView“ verwendet. Für verschiedene RS-Codes, insbesondere ATHLET, wurden Verfahren entwickelt, mit denen die Daten in ein für ParaView geeignetes Format übertragen werden kann. Die transformierten Daten enthalten neben der Beschreibung der Geometrie auch die dynamischen Ergebnisdaten sowie Informationen zum verwendeten Simulationsgitter. Die Erzeugung der Daten für ParaView wird durch eine Reihe von Python-Skripten und ein grafische Bedienoberfläche unterstützt.

Der erzeugte EXODUS II-Datensatz kann direkt in ParaView geöffnet werden. In der Darstellung der Daten gibt es dann sehr flexible Ansichten, die beispielsweise mit speziellen Filterfunktionen, der Ein-und Ausblendung von Objekten, Transparenz, Nutzung von Schnittebenen u.v.m. entsprechend der gewünschten Information angepasst werden können. Besonders nützlich ist die zusätzliche Darstellungsmöglichkeit der Flussrichtung des Kühlmittels durch 3D-Vektoren, die zum Verständnis der lokalen Effekte beiträgt. Die Visualisierung in ParaView ist vor allem bei der Verwendung sehr detaillierter Diskretisierung in den Codes im Vorteil gegenüber 2D-Bildern, wie sie in ATLAS möglich sind. Dies gilt besonders für die Analyse von Phänomenen mit starken, lokalen Unterschieden und die Validierung mehrdimensionaler Modelle.

9.1 CAD Werkzeuge zur 3D-Modellierung

Basierend auf dem frei verfügbaren CAD-System FreeCAD wurden zur Anlagenmodellierung verschiedene Vorgehensweisen und Techniken zur intuitiven Konstruktion von 3D-Geometrie entwickelt. Die hierbei besten und im Hinblick auf die Anwendung mit ATHLET-Eingabedaten erfolgversprechendsten Ansätze wurden weitestgehend automatisiert und innerhalb eines sog. *FreeCAD-Workspaces* zusammengefasst. Mit den zur Verfügung gestellten Befehlen können Rohrleitungsgeometrien intuitiv konstruiert werden. Anlagenpläne, welche oft in Form von Isometrieplänen vorliegen, liefern in der Regel alle benötigten Maße und Daten, die meist direkt und ohne manuelle Umrech-

nungen beim Aufruf der Befehle eingegeben werden können. Notwendige Berechnungen erfolgen automatisch und erzeugen unmittelbar die daraus entstehenden Rohrleitungsgeometrien. Dadurch kann die Korrektheit der eingegebenen Parameter, wie Längen und Winkel, direkt in der 3D-Ansicht überprüft werden. Armaturgeometrien können aus einfachen Konstruktionsschritten zusammengebaut werden. Häufig gebrauchte Armaturen, wie Reduzierungen, Ventile und Pumpen wurden bereits zusammengefasst und stehen im Workspace als eigene Befehle bereit. Der Benutzer kann die Konstruktionsbefehle über die interaktiven Bedienelemente, wie Buttons, dem Dokumentbaum und den Eingabedialogen zur Anpassung von Parametern, bedienen. Die Reihenfolge und TFO-Zuordnung der so erstellten Konstruktionsschritte können im Dokumentbaum per *Drag and Drop* angepasst werden. Neben der interaktiven Bedienung sind alle Befehle weiterhin über die Python-Konsole zugänglich, was es einfach macht neue Makrobauteile zu entwickeln und komplexe Konstruktionen per Skript zu erstellen.

Aufgrund ihrer Komplexität und der hohen Anzahl an notwendigen Parametern ist die Erstellung von RDB-Geometrien nur teilweise automatisierbar. Durch Unterteilung von RDBs in regelmäßige und symmetrische Teile ist es dennoch möglich, nodalisierte Geometrien schnell zu erstellen. Die besten Ergebnisse konnten durch Verwendung der in FreeCAD eingebauten Array-Befehle erzielt werden, deren Parameter auch nachträglich änderbar sind. Dadurch bleibt ein so erstelltes RDB-Modell in einem gewissen Rahmen flexibel und sollte in den meisten Fällen leicht an veränderte Bedingungen, wie Höhenänderungen, Nodeanzahl oder Kanalanzahl angepasst werden können.

Nach abgeschlossener Konstruktion kann die Exportfunktion benutzt werden um die Nodemeshes der erzeugten Geometrien für die weitere Verarbeitung im OBJ-Format abzuspeichern. Hier wird ein halbautomatisches Verfahren benutzt um die Meshes eindeutig und gemäß der ATHLET-Nomenklatur zu benennen.

Weitere Entwicklung sollte unternommen werden um die Stabilität und Kompatibilität der Workspace-Funktionen mit zukünftigen FreeCAD-Versionen sicherzustellen. Darüber hinaus sollte die interaktiven Bedienbarkeit einiger Konstruktionsbefehle überarbeitet werden. Gerade durch die rege Weiterentwicklung des FreeCAD-Basissystems eröffnen sich immer wieder neue Möglichkeiten Erweiterungen transparent einzubinden. Anwendern sollte ein einfacher Weg angeboten werden, den entwickelten Workspace als Paket in einer vorhandenen FreeCAD-Installation nachzurüsten.

9.2 Modellierung mit grafischen Netzwerkeditoren

Die grafische Eingabeerstellung für das „General Control System Model“ (GCSM) wird durch den „ATHLET GCSM Modeller“ (AGM) sehr gut unterstützt und kann bereits in der produktiven Anwendung, z. B. zur Modellierung komplexer leittechnischer Systeme, eingesetzt werden. Vorhandene umfangreiche GCSM-Modelle in AGM zu übertragen ist mit relativ hohem Aufwand verbunden und es können Abweichungen zu der vorhandenen Simulation auftreten, die analysiert werden müssen. Eine generelle Umstellung der vorliegenden Systemsimulationen auf Basis von AGM ist daher gegenwärtig nicht sinnvoll. Für die Modellierung neuer Systeme wird hingegen empfohlen, ausschließlich AGM einzusetzen

Es wird vorgeschlagen AGM in zweierlei Hinsicht zu verbessern. Die vorhandenen, umfangreichen Leittechnikbibliotheken in den verschiedenen Anlagensimulatoren für deutsche KKW sind überwiegend in dem prioritären G2-Modeler /JAK 97/ erstellt worden. Mit der Entwicklung eines Konverters wäre es möglich, diese Daten weitgehend automatisch für AGM zu übernehmen und damit die vorhandene Leittechniksimulation für die weitere Bearbeitung, Dokumentation oder als Basis für ähnliche Systeme bereitzustellen. Generell wird eine weitgehende Unabhängigkeit der interaktiven Werkzeuge vom verwendeten Betriebssystem angestrebt. Der Hersteller der Basissoftware bietet mittlerweile eine Version für die Linux Plattform an. Diese sollte daher beschafft und angepasst werden. um AGM auch unter diesem Betriebssystem verwenden zu können.

Eine Modellierung von Fluidsystemen mit ATHLET Thermo-Fluidobjekten (TFO) ist im Prototyp von ATM bereits im beschriebenen Umfang möglich. Damit konnte die prinzipielle Eignung der Basissoftware SimInTech für diese Anwendung gezeigt werden. Der Prototyp ist für den geplanten Einsatz, die vollständige interaktive Unterstützung der Eingabedatenerstellung von ATHLET, noch nicht einsatzfähig. Einerseits fehlt die Unterstützung zur Datenerstellung für viele Modelle und Optionen. Andererseits haben Diskussionen mit ATHLET Entwicklern und Anwendern ergeben, dass auch der gegenwärtige Zustand, insbesondere hinsichtlich der Darstellung der Geometrie und der Fehlerkontrolle, noch der Überarbeitung bedarf.

Für die weitere Entwicklung wird daher vorgeschlagen, den Prototyp auszubauen, fehlende Funktionen zu ergänzen und die Diskretisierung der TFO besser zu unterstützen. Wichtige Punkte sind beispielweise eine geometriegetreue Darstellung für die einzel-

nen TFOs und die grafische Anzeige der Zonengrenzen, ähnlich wie in der „ATHLET Input Graphic“. Zur realistischen Nachbildung der Anlage ist die Berücksichtigung wärmeleitender Strukturen notwendig. ATHLET nutzt dafür sogenannte Wärmeleitobjekte, die „Heat Conduction Objects“ (HCO). Eine Vielzahl von Daten, wie verwendete Korrelationen, Materialien, Art der Objekte, Geometrie, Diskretisierung und die Kopplung an die Fluidobjekte muss in der Eingabe definiert werden. In ATM sollen diese HCO implementiert werden und bereits bei der Erstellung eines HCO optional ein gekoppelte TFOs erzeugt werden können.

Zusätzlich wäre es nützlich, Daten automatisch kopieren zu können bzw. sich auf Referenzobjekte beziehen zu können, um den Aufwand für den Benutzer zu reduzieren. Die gegenwärtige Erweiterung von ATHLET mit einem 3D-Thermohydraulikmodul erfordert eine sehr umfangreiche Diskretisierung, die sehr zeitaufwändig und fehleranfällig ist und bei hoher Auflösung manuell nicht mehr durchführbar ist. Eine automatisierte Eingabedatenerzeugung in ATM für diese Modellierung, beispielsweise durch die Bereitstellung parametrischer Makroobjekte, wäre notwendig und wünschenswert.

Die Erweiterung von ATM sollte mit Maßnahmen der Qualitätssicherung ergänzt werden, die am Ende der Entwicklung ein System garantieren, das zuverlässig in der täglichen Anwendung eingesetzt werden kann. Entsprechend dem QM-Handbuch der GRS /GRS 13/ gehören dazu z. B. eine enge Abstimmung mit dem ATHLET-Entwicklungsteam, Dokumentation der Programme, ein User Manual sowie umfangreiche Tests für verschiedene Anlagendatensätze.

9.3 3D-Visualisierung von Simulationsdaten

In dem Projekt wurde eine 3D-Visualisierungslösung für ATHLET und Programme der Neutronendynamik auf der Basis von freien Datenformaten und quelloffenen Programmen geschaffen. Die enge Zusammenarbeit mit Anwendern und frühe Anwendungstests stellten dabei sicher, dass den speziellen Anforderungen des Programms und seiner Benutzer Rechnung getragen werden konnte. Wegen struktureller Ähnlichkeiten sind die gefundenen Lösungsansätze und geschriebenen Werkzeuge möglicherweise mit relativ geringen Anpassungen auf andere 1D-Codes der GRS wie COCOSYS und Simulationsprogramme aus dem Bereich Endlagerung übertragbar.

Weitere Entwicklungsarbeiten sind erforderlich, um eine bessere Skalierung der Visualisierungslösung in Bezug auf die Anzahl der verwendeten Objekte zu erreichen. Expe-

rimentelle ATHLET-Datensätze definieren zum Teil mehrere zehntausend Objekte. Hier stößt das NetCDF3-Format an seine Grenzen. Das alternativ verwendbare NetCDF4, das als Backend HDF5 verwendet, kann theoretisch mit großen Objektmengen umgehen, litt aber in ersten Tests an inakzeptabel langsamen Zugriffsgeschwindigkeiten, die möglicherweise von der Kompatibilitätsschicht zwischen NetCDF und HDF5 herrühren.

Verbesserungswürdig ist des Weiteren der Datenaustausch zwischen FreeCAD und obj2exii. Einige Informationen, die bei der Erstellung der Geometrie schon vorliegen, wie z. B. die Lage und Richtung der 1D-Vektoren oder die Position von Objektzentren, ließen sich als Metadaten in der OBJ-Datei speichern. Zurzeit kommt es bei geometrisch komplexeren, z. B. stark gekrümmten oder nicht-konvexen Objekten zu Artefakten in der Darstellung von Flussgrößen, weil aufgrund von Informationsmangel mit geometrischen Approximationen gearbeitet wird.

Die Integration von 1D-Flussgrößen aus ATHLET zu 3D-Vektoren hat sich als sehr nützliches Hilfsmittel erwiesen, um qualitativ die objektübergreifenden Strömungseigenschaften eines Modells visuell zu erfassen. Es muss jedoch deutlich gemacht werden, dass die intuitive, aber leider auch naive Interpretation der Vektoren in der Nähe geometrisch komplexer Objekte oder Wände nicht erlaubt ist. Möglicherweise existieren bessere Methoden zur Addition der Flüsse eines Objekts, die an diesen Stellen sinnvollere Resultate liefern.

9.4 Analysesystem ATLAS

Der vorhandene Funktionsumfang von ATLAS für das Postprozessing von ATHLET und anderen RS-Codes ist sehr groß und deckt die wichtigsten Anforderungen ab. Die Visualisierungsmöglichkeiten sind sehr umfangreich, z. B. X/Y-Diagramme, komplexe 2D-Prozessbildern oder vereinfachte 3D-Darstellungen. Einige Diagramme, wie die Visualisierung von ATHLET-Objektnetzwerken, für das Brennstabverhalten oder Dreistoffdiagramme für H_2-Verbrennung können automatisiert erzeugt werden und erfordern nur geringen Aufwand bei der Bilderstellung.

Im Gegensatz dazu entspricht die Implementierung des Programms nicht mehr dem Stand der aktuellen Computertechnik. Der Programmcode ist durch das kontinuierliche Hinzufügen neuer Funktionen zunehmend unübersichtlich und in den Details schwer verständlich geworden, auch wegen der unzureichenden Dokumentation der Erweite-

rungen. Dadurch sind die Programmpflege und das Hinzufügen neuer Funktionalität nur mit großem Aufwand durchführbar. Der Code ist außerdem an proprietäre Bibliotheken gebunden, die die Anwendbarkeit auf Windows beschränken. Die gleichzeitige Verwendung von Programmiersprachen C++ und FORTRAN verstärkt die genannten Probleme.

Zur Lösung der beschriebenen Probleme ist substantielles Reengineering der Software nötig. Damit kann die langfristige Verfügbarkeit und Erweiterbarkeit von ATLAS sichergestellt werden. Für die weitere Entwicklung ist daher das Hauptziel die Softwarequalität zu steigern. Gleichzeitig sollte zusätzlich die bisher fehlende Lauffähigkeit unter Linux realisiert werden. Alle Erweiterungen sollten möglichst unter Nutzung quelloffener Software („Open Source“) umgesetzt werden, um von deren weiterer Entwicklung zu profitieren und gleichzeitig eine lizenzkostenfreie Lösung zur Verfügung zu stellen.

Im Einzelnen wären folgende Maßnahmen im Programm sinnvoll:

- die Modularisierung und strukturelle Verbesserung (Refactoring) des Codes
- die Erweiterung der Programmdokumentation,
- Ersatz des Moduls für die Erstellung von X/Y-Diagrammen,
- Ersatz der GUI-Bibliothek durch Open Source-Varianten.
- die Standardisierung des ATLAS Bildformats.

Literatur

/ALL 05/ Allelein, H.-J., et al.
Weiterentwicklung der Rechenprogramme COCOSYS und ASTEC, Abschlussbericht, GRS-A-3266, 2005.

/APT 07/ Advanced Visual Systems Inc., April 2007, Symbolic Nuclear Analysis Package (SNAP), User's Manual.

/APT 15/ Applied Programming Technology Inc., AptPlot - A Free Pure-Java 2D Plotting Tool, Bearbeitungsstand: 15. Februar 2015, 13:00.,
URL: https://www.aptplot.com/services/index.jsp.

/ATH 09/ ATHLET Mod 2.2 Cycle A, 2009, Code Documentation Package, GRS-P-1: Vol. 1: User's Manual, Vol. 2 : Programmer's Manual, Vol. 3 : Validation, Vol. 4 : Models and Methods.

/AUS 90/ Austregesilo, H., Voggenberger, T., 1990, The General Control System Module GCSM within the ATHLET Code, GRS-A-1660.

/AVS 15/ Applied Programming Technology Inc., AVS Express, Bearbeitungsstand: 15. Februar 2015, 13:00., URL: http://www.avs.com/solutions/express/.

/CHR 10/ Christienne, M. et al., 2010, COUPLED TORT-TD/CTF CAPABILITY FOR HIGH-FIDELITY LWR CORE CALCULATIONS, PHYSOR 2010– Advances in Reactor Physics to Power the Nuclear Renaissance, Pittsburgh, Pennsylvania, USA

/DEI 14/ Deitenbeck, H., 2014, ATHLET-GCSM Modeller (AGM) User's Manual.

/DEI 15/ Deitenbeck, H., 2015, AGM Programmdokumentation für Entwickler.

/EMB 15/ Embarcadero Technologies, Delphi XE 7, Bearbeitungsstand: 15. Februar 2015, 13:00., URL: http://www.embarcadero.com/de/products/delphi.

/EXO 95/ Larry A. Schoof, Victor R. Yarberry: „EXODUS II: A Finite Element Data Model"; Computational Mechanics and Visualization Department; Sandia National Laboratories; Albuquerque, NM 87185 (1995).

/FRC 15/ FreeCAD, Open Source parametric 3D CAD modeler, Bearbeitungsstand: 15. Februar 2015, 13:00., URL: http://www.embarcadero.com/de/products/delphi.

/FZD 15/ Forschungszentrum Dresden Rossendorf, DYN3D-ATHLET, Computer Model for Steady-State and Transient Analysis of Nuclear Reactor Cores, Bearbeitungsstand: 03. Februar 2010, 13:00., URL: http://www.hzdr.de/db/Cms?pOid=11771&pNid=2737.

/GRS 13/ Ges. für Anlagen- und Reaktorsicherheit (GRS) mbH, Maßnahmen zur Qualitätssicherung bei der Erstellung von Computerprogrammen in der GRS (QM-Richtlinie Programmentwicklung), QM-Handbuch, Nov. 2013.

/GRS 15/ Ges. für Anlagen- und Reaktorsicherheit (GRS) mbH, Qualitätsmanagement, Bearbeitungsstand: 15. Februar 2015, URL: http://www.grs.de/content/qualitaetsmanagement/.

/HDF 97/ The HDF Group. Hierarchical Data Format, version 5, 1997-2015. http://www.hdfgroup.org/HDF5/.

/INT 15/ Intelligent Light, FieldView 15, Bearbeitungsstand: 15. Februar 2015, 13:00., URL: http://www.ilight.com/en/products/fieldview-15.

/JAK 97/ Jakubowski, Z.,
Leitechnikmodule des Reaktorschutz-Reaktorleistungsbegrenzungssystems und der elektrischen Stromversorgung für das KKW Neckar II (GKN II), GRS-A-2542, 1997.

/KLI 09/ Kliem, S.; et al.,
Integration of DYN3D into the NURESIM Platform and Coupling with FLICA, NURISP General Seminar, FZD Dresden, 1997.

/LAN 05/ Langenbuch, S., Velkov, K., 2005, Overview on the Development and Application of the Coupled Code System ATHLET-QUABOX/CUBBOX, M & C + SNA05– Mathematics and Computation, Supercomputing, Reactor Physics and Nuclear and Biological Applications, Avignon, France.

/LER 12/ Lerchl, G.; et al., ATHLET Mod 3.0 Cycle A User's Manual, GRS-P-1/Vol. 1 Rev. 6, 2012.

/LLN 15/ Lawrence Livermore National Laboratory, VisIt - An Open Source, interactive, scalable, visualization, animation and analysis tool, Bearbeitungsstand: 10. Februar 2015, 13:00., URL: https://wci.llnl.gov/simulation/computer-codes/visit/.

/MAT 15/ The MathWorks Inc., Simulink, Simulation und Model-Based Design, Bearbeitungsstand: 10. Februar 2015, 13:00., URL: http://de.mathworks.com/products/simulink.

/MEY 15/ Prof. Dr.-Ing. Dagmar Meyer, Ostfalia Hochschule für angewandte Wissenschaften, Institut für Elektrische Anlagen und Automatisierungstechnik, Bearbeitungsstand: 10. Februar 2015, 13:00., URL: http://www.ostfalia.de/export/sites/default/de/pws/meyer/lehrveranstaltungen/rtlabor/skripte/LabRT_V5_2_0.pdf.

/NET 90/ Rew, R. K., Davis. G. P.: „NetCDF: An Interface for Scientific Data Access“; IEEE Computer Graphics and Applications, Vol. 10, No. 4, pp. 76-82 (1990) http://www.unidata.ucar.edu/software/netcdf/.

/OBJ 89/ Wavefront Technologies: OBJ http://www.martinreddy.net/gfx/3d/OBJ.spec.

/OCT 15/ Open CASCADE Technology, Bearbeitungsstand: 10. Februar 2015, 13:00., URL: http://www.opencascade.org/.

/PAP 09/ Papukchiev, A. , Lerchl, G. et al., Sept. 27- Oct.3, 2009, Extension of the Simulation Capabilities of the 1D System Code ATHLET by Coupling with the 3D Software Package ANSYS CFX; Proceedings of NURETH-13 Conference, Kanazawa, Japan.

/PAR 15/ Utkarsh Ayachit: „The ParaView Guide: A Parallel Visualization Application". Kitware Inc., Clifton Park, NY (2015) http://www.paraview.org.

/PER 10/ Périn Y., Velkov K., Pautz A., 2010, COBRA-TF / QUABOX-CUBBOX: CODE SYSTEM FOR COUPLED CORE AND SUBCHANNEL ANALYSIS, PHYSOR 2010– Advances in Reactor Physics to Power the Nuclear Renaissance, Pittsburgh, Pennsylvania, USA.

/POI 15/ Pointwise Inc., Gridgen - Reliable CFD Meshing, Bearbeitungsstand: 10. Februar 2015, 13:00., URL: http://www.pointwise.com/gridgen/.

/PYT 15/ Python Software Foundation: "Python 2.7.9 documentation" (2015) https://docs.python.org/2/.

/SAL 15/ SALOME, Open Cascade S.A, Bearbeitungsstand: 10. Februar 2015, 13:00., URL: http://www.salome-platform.org/.

/SCI 15/ Scilab Enterprises, Scilab - Open source software for numerical computation,Bearbeitungsstand: 10. Februar 2015, 13:00., URL: http://www.scilab.org/.

/SEU 08/ Seubert, A., Velkov, K., Langenbuch, S., 2008, The Time-Dependent 3-D Discrete Ordinates Code TORT-TD with Thermal-Hydraulic Feedback by ATHLET Model, PHYSOR 2008, Interlaken, Switzerland.

/SIT 15A/ 3V Services, SimInTech, Bearbeitungsstand: 10. Februar 2015, 13:00., URL:http://3vservices.com/ru/index.php?option=com_content&view=article&id=68&Itemid=69.

/SIT 15B/ 3V Services, 2015, SimInTech User Manual, General information on SimInTech.

/TET 15/ Hang Si: "TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator" ACM Trans. on Mathematical Software. 41 (2), Article 11 (2015); http://wias-berlin.de/software/tetgen/

/TRA 15/ Trac Open Source Project, Bearbeitungsstand: 10. Februar 2015, 13:00., URL: http://trac.edgewall.org/.

/VOG 10/ Voggenberger, T., 2010, ATLAS Vista: Ausbau der Ergebnisdarstellung und Datenverwaltung, GRS-A-3577.

/VOG 14A/ Voggenberger, T. et al., 2014, ATLAS 5.1 User's Manual V 1.0.

/VOG 14B/ Voggenberger, T. et al., 2014, APG 5.1 User's Manual V 1.0.

/WIK 15/ Volume rendering, Bearbeitungsstand: 15. Februar 2015, 13:00., URL: http://en.wikipedia.org/wiki/Volume_rendering.

Abbildungsverzeichnis

A.1 Funktionen der CAD-Systeme im Vergleich

2D Features	Naro CAD 1.6.2 Alpha	Free CAD	Google SketchUp 8	Turbo CAD 18 Prof.	Ansys 12.1
Skizze Modus					
Auf normalisierte Ebene	✓	✓	✗	✓	✓
Auf einem Körper	✗	✓	✗	✓	
Auf eine eingefügte Ebene	✓	✓	✗	✓	✓
Intuitive Skizze	✗	✗	✗	✗	✓
Einfache 2D Geometrie					
(Linien, Rechteck, Bögen, Kreise, Drähte, etc)	✓	✓	✓	✓	✓
Text eingeben	✓	✓	✗	✓	✓
Maße eingeben	✓	✓	✓	✓	✓
Maße parametrisieren	✗	✓	✗	✓	✓
Intuitive Messung	✗	✗	✗	✗	✓
Graphische Modifikationen von Operationen					
Translation	✓	✓	✗	✓	✓
Rotation	✓	✓	✗	✓	✓
Scaling	✓	✓	✓	✓	✓
Spiegel Funktion	✓	✓	✗	✓	✓
Matrix	✓	✓	✗	✓	✓
Intersection	✓	✓	✓	✓	✓
Substraction	✓	✓	✓	✓	✓
Import	✓	✓	✓	✓	✓
Export	✓	✓	✓	✓	✓

3D Features	Naro CAD 1.6.2 Alpha	Free CAD	Google SketchUp 8	Turbo CAD 18 Prof.	Ansys 12.1
Einfache Körper					
Würfel	✓	✓	✓	✓	✓
Kugel	✓	✓	✗	✓	✓
Zylinder	✓	✓	✗	✓	✓
Kegel u.s.w.	✓	✓	✗	✓	✓
Körper erzeugen					
Extrudieren	✓	✓	✓	✓	✓
Cut	✗	✓	✓	✓	✓
Rotation	✓	✓	✗	✓	✓
aus Flächen	✗	✗	✓	✓	✓
Shape und Guideline	✓	✗	✗	✗	✓
Spiegel Funktion	✓	✓	✗	✓	✓
Volumen parametrisieren					
Boolsche Operationen					
Addieren	✓	✓	✓	✓	✓
Schneiden	✓	✗	✓	✓	✓
Intersection	✓	✓	✓	✓	✓
Verbinden	✓	✓	✓	✓	✓
Differenz	✓	✓	✓	✓	✓
Matrix Funktion	✓	✓	✓	✓	✓
Farben und Texturen	✓	✓	✓	✓	✓
Stoffe und Dichte	✗	✗	✗	✗	✓

Allgemeines	Naro CAD 1.6.2 Alpha	Free CAD	Google SketchUp 8	Turbo CAD 18 Prof.	Ansys 12.1
Parallele Ebene nach XYZ erstellen	✓	✓	✗	✓	✓
Nicht parallele Ebene erstellen	✗	✗	✗	✗	✗
Ebene auf einem Körper erstellen	✗	✗	✗	✓	✓
Queransicht auf eine Ebene	✗	✗	✗	✓	✓
Direkt Scripting Möglichkeiten	LUA	PYTHON	RUBY	✗	✗
Formate zum Exportieren	.naroxml	.agdb, .x_t, .x_b, .anf, .igs, .stp, .mcnp, etc.	.dxf, .dwg	.3ds, .dgn, .dwg, .dxf, .igs, .mtx, .sat, shx, .skp, .step, .svg, .wmf, etc	Haupt CAD-Formate
Formate zum Importieren		.stl, .ast, .obj, .nas, .iv, .bms	.3ds, .dxf, .dwg	.dwg, .dxf, .jpeg, .bmp, PICT, Spline, Claris CAD	.iges, .step, .sat + Haupt CAD-Formate

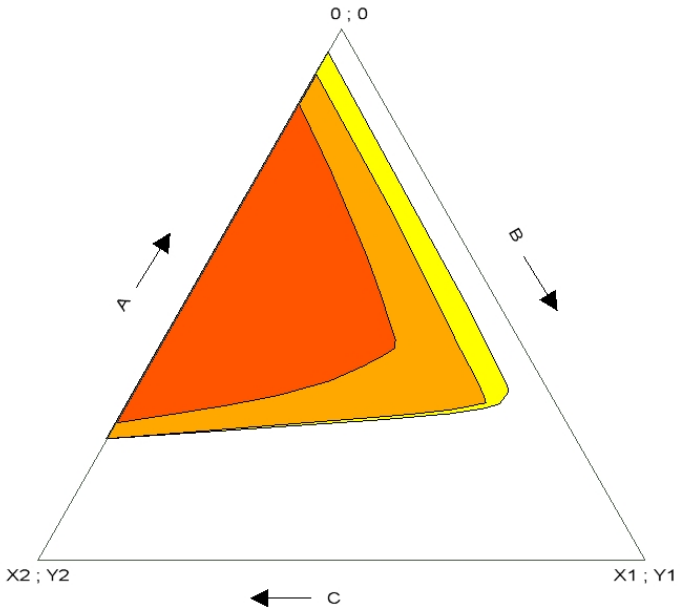
A.2 Berechnungen für das Shapiro Diagramm

Koordinatenberechnung des darstellenden Punkts des Stoffgemisches.

Wichtige Voraussetzung für die voll automatische Erzeugung eines Dreistoffdiagramms ist es, in Abhängigkeit zu den unterschiedlichen Codeergebnissen, die korrekte Positionsberechnung des darstellenden Punktes des Stoffgemisches.

Dazu werden in diesem Anhang zuerst die allgemeinen Methoden zur Berechnung von Dreistoffdiagrammen erläutert.

Gegeben sind die relativen Konzentrationen A, B und C von Stofftypen mit Grenzwerten 0 und 1 in einem beliebigen Raum. Wir wollen in Bezug auf das hier unten abgebildete Dreistoffdiagramm die Koordinaten X und Y vom darstellenden Punkt des Stoffgemisches berechnen.



Als notwendige Festlegungen für die Berechnung gelten folgende Annahmen:

- Die Summe der Stoffkonzentrationen ist gleich Eins.
- Gemäß dem orthogonalen Bezugsystem, stehen die Punktkoordinaten in linearer Abhängigkeit zu den Konzentrationen A, B und C.

Also:

1. A + B + C = 1
2. X = ax*A + bx*B + cx*C
3. Y = ay*A + by*B + cy*C

Wobei die Koeffizienten ax, bx, cx, ay, by und cy zu berechnen sind.

Unter Berücksichtigung der im Diagramm gegebenen Randbedingungen muss für

B = 1 folgendes gelten: A = 0, C = 0, X = X1 und Y = Y1.

Daraus folgt:

bx = X1 und by = Y1

Die analoge Berechnung für C = 1 und A = 1 ergeben:

ax = 0, ay = 0, cx = X2 und cy = Y2.

Daher folgt:

X = X1*B + X2*C
Y = Y1*B + Y2*C

Aus diesen Gleichungen lässt sich ableiten, dass die gesuchten Punktkoordinaten ausschließlich von den Stoffkonzentrationen und von der Geometrie des Shapiro-Dreiecks abhängen.

Rechenprogrammorientierte Berechnung von Konzentrationen

Die Konzentrationen der Stoffe im Shapiro-Diagramm werden in Volumenanteile angegeben. Für die geometrische Berechnung der Punktposition im Bild ist es dann notwendig, wie bereits erwähnt, dass die Summe aller Stoffanteile gleich 1 sei. Aus den für Testzwecke zur Verfügung stehenden Daten lässt sich feststellen, dass für die Stoffkonzentrationen der drei in Betracht kommenden Rechenprogramme jeweils unterschiedliche Umrechnungen notwendig sind.

In MELCOR werden die Konzentrationen in Volumenanteile mit zwischen 0 und 1 variierenden Werten angegeben, deswegen ist nur die Umrechnung nötig, die aus dem Sauerstoffanteil die Konzentration des in der Luft vorhandenen Stickstoffes berechnet.

IN COCOSYS werden die Konzentrationen der Volumenanteile in Prozent angegeben, deswegen ist für die Umrechnung die Division aller Konzentrationsanteile durch 100 notwendig.

Komplexer, ist die Umrechnung für ASTEC. Aus der aus mehreren Rechnungen zur Verfügung stehenden Daten lässt sich Folgendes feststellen:

1. Für den Wasserdampf steht nur der Partialdruck zur Verfügung. Davon lässt sich dann der Volumenanteil des Wasserdampfs mittels Division durch den gesamten Druck berechnen.
2. Für alle anderen Stoffe stehen die Massen zur Verfügung. Diese müssen deswegen zuerst durch die für jeden Stoff charakteristische molare Masse (32 für O_2, 28 für N_2 und CO, 2 für H_2 und 44 für CO_2) dividiert werden. Dann, zur Berechnung der Volumenanteile werden diese einzelnen Werte noch durch ihre Summe geteilt. Zuletzt müssen sie noch mit dem Restanteil (1.0 – Dampfanteil) multipliziert werden.

Dabei ist es zu beachten, dass, bei all diesen Berechnungen, für die betreffenden Stoffe die Eigenschaften idealer Gase vorausgesetzt werden.

Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH

Schwertnergasse 1
50667 Köln
Telefon +49 221 2068-0
Telefax +49 221 2068-888

Forschungszentrum
85748 Garching b. München
Telefon +49 89 32004-0
Telefax +49 89 32004-300

Kurfürstendamm 200
10719 Berlin
Telefon +49 30 88589-0
Telefax +49 30 88589-111

Theodor-Heuss-Straße 4
38122 Braunschweig
Telefon +49 531 8012-0
Telefax +49 531 8012-200

www.grs.de

ISBN 978-3-944161-49-5